

Eliisa Niskanen

ADOBE EDGE ANIMATE

Askel kohti tulevaisuuden bannerituotantoa

ADOBE EDGE ANIMATE

Askel kohti tulevaisuuden bannerituotantoa

Eliisa Niskanen
Opinnäytetyö
Syksy 2014
Tietojenkäsittelyn koulutusohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietojenkäsittelyn koulutusohjelma
Digitaalinen media ja internetpalvelut

Tekijä: Eliisa Niskanen

Opinnäytetyön nimi: Adobe Edge Animate – Askel kohti tulevaisuuden bannerituotantoa

Työn ohjaaja: Pekka Ojala

Työn valmistumislukukausi- ja vuosi: syksy 2014

Sivumäärä: 44 + 4

Opinnäytetyössä kerrotaan Adobe Systemsin kehittämästä, HTML5-animaatioiden tekoon tarkoitettua ohjelmasta nimeltään Adobe Edge Animate. Toimeksiantajana toimii Kaleva Oy. Opinnäytetyössä selvitetään, kuinka ohjelma soveltuu lehtitalon tuleviin tarpeisiin ja sulautuu jo olemassa oleviin järjestelmiin.

Opinnäytetyön teoriassa kerrotaan ohjelman käyttämistä teknologioista sekä sen tärkeimmät toiminnot kehitystehtävien kannalta. Opinnäytetyön käytännön osuudessa toteutettiin ohjelmalla neljä erilaista display- eli banneripohjaa, joista kukin sisälsi erilaisia toiminnallisuksia.

Ohjelma hyödyntää toiminnassaan HTML5-kieltä, CSS3-tyyliohjekieltä sekä JavaScriptiä, eritoten jQuery-kieltä. Ohjelman paras puoli onkin se, että sillä voi tehdä helposti näyttävän näköisiä bannereita riippumatta käyttäjän ohjelmointitaidoista. Animate nimittäin luo kaiken ohjelmakoodin työskentelyn taustalla. Ohjelman edistyneempi käyttö vaatii käyttäjältä kuitenkin ohjelmointiosaamista ja ohjelman käyttämien teknologioiden tuntemusta.

Haasteeksi muodostui HTML5-kielen ja CSS3-tyyliohjekielen selaintuki. Kumpaakaan ei ole vielä virallisesti standardisoitu, vaikka uusimmat selaimet niiden piirteitä jo kiitettävästi tukevatkin. Kaikkinensa Adobe Edge Animate soveltuu loistavasti lehtitalon tuleviin tarpeisiin. Ohjelma on uusi Kalevan olemassa olevalle verkkomainonnan järjestelmälle, joten aineistonsiirto on toistaiseksi hieman monimutkaista. Erikoisbannereiden kohdalla toimeksiantajan kannattaa tutustua tarjolla oleviin, moderneja web-teknologioita hyödyntäviin verkkomainonnan toimijoihin, joilla on tarjota myös erikoisempia banneriratkaisuja Animatelle toteutettujen bannereiden rinnalle.

Asiasanat: Adobe Edge Animate, HTML5, CSS3, JavaScript, jQuery, display-mainonta

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Business Information Systems,
Option of Internet Services and Digital Media

Author(s): Eliisa Niskanen

Title of Bachelor's thesis: Adobe Edge Animate – Tool to Display Advertising in Future

Supervisor: Pekka Ojala

Term and year of completion: Autumn 2014

Number of pages: 44 + 4

This thesis describes a program called Adobe Edge Animate, which was created by Adobe Systems to generate HTML5 animations. The thesis was commissioned by Kaleva PLC. The thesis will determine how well Adobe Edge Animate is suited for the future needs of the company and how well it integrates with the existing programs.

The theoretical framework clarifies the technologies behind the program and its most important features regarding this development task. Four different display templates, aka banner templates, with different functionalities were created as examples for the commissioner.

The program utilizes HTML5 language, CSS3 stylesheet language and JavaScript, especially jQuery language. One of the best features of the program is that it can be used without prior knowledge of computer programming languages. Adobe Edge Animate will create all the necessary coding in the background. A more advanced use requires prior knowledge of programming and used technologies.

The HTML5 language and CSS3 stylesheet language are not yet fully standardized, and this can be seen as the biggest challenge for the future use of the program within Kaleva PLC, despite the fact that most new browsers offer adequate support already. In conclusion, it can be said that Adobe Edge Animate is suitable for the future needs of the commissioner.

Adobe Edge Animate is a new tool within the existing management tools of Kaleva's online systems and that is why transferring materials between the different programs is rather complicated at the moment. The most advanced banners could be outsourced to the companies which specialize in modern web technologies and can offer special banners in addition to those created with Adobe Edge Animate.

Keywords: Adobe Edge Animate, HTML5, CSS3, JavaScript, jQuery, display advertising

SISÄLLYS

1	JOHDANTO	6
2	MODERNIT WEB-TEKNOLOGIAT	7
2.1	HTML5	7
2.2	CSS3	9
2.3	JavaScript	11
2.4	jQuery	11
3	ADOBE EDGE ANIMATE CC	13
3.1	HTML ja Edge Animate	19
3.2	CSS ja Edge Animate	22
3.3	JavaScript ja Edge Animate	24
4	DISPLAY- ELI BANNERIMAINONTA	28
4.1	Display-mainonta Kaleva Oy:ssa	29
4.2	Display-mainonnan tulevaisuus Kaleva Oy:n näkökulmasta	31
5	BANNERIPOHJIEN TOTEUTUS	32
5.1	Haitaribanneri	33
5.2	Slider	34
5.3	3D-kuutiokaruselli	36
5.4	Videobanneri	38
5.5	Aineistonsiirto	39
6	TULOKSET	40
7	POHDINTA	41
	LÄHTEET	42
	LIITTEET	45

1 JOHDANTO

Ihmiset käyttävät mobiililaitteita yhä enemmän ja enemmän. Mainostajallakin on tärkeää pysyä tämän kehityksen mukana. Mainonta ei ole enää pelkkää perinteistä printtimainontaa, vaan se suuntautuu yhä enemmän verkkoon.

Tulee heti mieleen muutama tulevaisuuden kannalta merkittävä teknologia, jotka vastaavat tähän tarpeeseen. Ne ovat nimeltään HTML5, CSS3 ja JavaScript. Näiden teknologioiden avulla voidaan toteuttaa selaimesta ja laitteesta riippumattomia ratkaisuja jokaisen käyttäjän ulottuville. Monet sanovat HTML5-kielen olevan vasta tulossa, mutta itseasiassa HTML5 on jo täällä.

Suuntautumisvaihtoehdon innoittamana ja aiemman osaamisen pohjalta keksittiin opinnäytetyöaihe, joka tutustuttaa tähän uuteen, mielenkiintoiseen teknologiaan ja josta ei monta opinnäytetyötä ole vielä kirjoitettu. Opinnäytetyössä kerrotaan Adobe'n uudesta, HTML5-animaatioiden tekoon tarkoitettusta ohjelmasta nimeltä Adobe Edge Animate. Toimeksiantajana toimi Kaleva Oy. Opinnäytetyössä selvitettiin, kuinka ohjelma soveltuu lehtitalon tuleviin tarpeisiin ja sulautuu jo olemassa oleviin järjestelmiin.

Työssä toteutettiin Adobe Edge Animatella neljä erilaista banneripohjaa, joista kaikki olivat toiminnallisuuksiltaan hieman erilaisia.

2 MODERNIT WEB-TEKNOLOGIAT

Moderneista web-teknologioista puhuttaessa mainitaan usein termit HTML5, CSS3 ja JavaScript. Yhdistettynä nämä kolme teknologiaa mahdollistavat paljon. Kyseiset teknologiat ovat frontend-teknologioita. Frontend tarkoittaa selaimen tulkitsemää ja käyttäjille visuaalisessa muodossa esitettyä osaa internet-sivustosta. HTML vastaa sivuston sisällöstä ja rakenteesta, CSS hoitaa sivuston visuaalisen tyylin määrittelyn sekä JavaScript luo sivustolle dynaamisia toimintoja. (Konsepti 2014, Viitattu 5.10.2014.) Tässä työssä frontend-tekniikoina esitellään nämä kolme teknologiaa sekä omana osuutenaan jQuery, joka on yksi suosituimpia JavaScript-kirjastoja.

2.1 HTML5

Termi HTML5 on monitahoinen, koska se tarkoittaa eri yhteyksissä erilaisia asioita. Korpela, joka on tutkinut asiaa, kertoo kirjassaan (2014, 3), että HTML5 voi tarkoittaa HTML-kielen uutta kehittyneempää versiota, yleisnimitystä nykyaikaisille web-teknologioille tai sovellusten toteuttamista web-teknologioilla. Tässä työssä HTML5 esitellään HTML-kielen uutena versiona.

Web-konsortio (W3C), eli World Wide Web Consortiumin vuonna 2004 järjestämässä kokouksessa ehdotettiin HTML-kielen laajentamista HTML 4:n ja selainten pohjalle. Tällöin ehdotus hylättiin, koska enemmistö kannatti HTML:n kehittämistä täysin uudelle pohjalle. Kehitystyö jakaantui kahdelle eri linjalle kannattavalle kehittäjälle. W3C:lle, joka halusi täysin uutta ja Applen, Mozillan ja Operan yhteistyön perustamalle Web Hypertext Application Technology Working Groupille (WHATWG), joka ryhtyi kehittämään kieltä laajentamalla olemassa olevaa HTML-kieltä. (Korpela 2014, 29.)

Nämä kaksi kehittäjää yhdistivät voimansa vuonna 2006, kun W3C ilmoitti edistyneensä kehitystyössä ja tulevaisuuden työskentelemään yhdessä WHATWG:n kanssa. Vuonna 2008 W3C julkaisi ensimmäisen HTML5-luonnoksensa. Yhteistyöstä huolimatta kehittäjillä on ollut erimielisyytensä ja niinpä he ovat jatkaneet HTML-kielen kehitystä omilla tahoillaan. (Korpela 2014, 29.)

Korpelan mukaan, WHATWG kokee HTML:n jatkuvan kehittämisen kutsumuksenaan ja näin ollen se on luopunut työnimestä HTML5 ja käyttää kehitystyön nimenä ”elävää HTML:ää” (”Living HTML Standard”), koska sen sisältö muuttuu hieman päivittäin. W3C sen sijaan toimii yritysten ja organisaatioiden muodostama konsortiona, jolloin sillä on oma monivaiheinen määrittelyjärjestelmänsä. Määritelmän kehitystyö voidaan jakaa viiteen luokkaan, jota kuvastaa taulukko 1. (2014, 29.)

TAULUKKO 1. W3C:n määrittelyjen luokitus valmiusasteen mukaan

WD, Working Draft	Työryhmän työversio. Muutokset todennäköisiä.
LC, Last Call	Viimeistely vaiheessa oleva työversio.
CR, Candidate Recommendation	Valmistellaan suositusehdotukseksi.
PR, Proposed Recommendation	SuosituSEhdotus, odottaa vahvistusta.
REC, Recommendation	Suositus, ”W3C:n standardi”.

W3C määritteli HTML5:n syksyllä 2013 tasolle CR, joka on lähestulkoon standardina pidetty. Määrittely pohjautui WHATWG:n määrittelyyn ja W3C:n tavoitteena on virallistaa HTML5 tämän pohjalta ilman suurempia muutoksia. WHATWG:n ja W3C:n kehitystyö muistuttavat paljon toisiaan, mutta näkemyserojen vuoksi ei ole löydetty täysin yhteistä linjaa HTML5:n määrittelylle. (Korpela 2014, 29–30.)

Korpela kertoo, että helmikuussa 2011 W3C kertoi tavoitteekseen julkaista HTML5:stä suositus eli W3C:n virallinen standardi vuoden 2014 viimeisellä neljänneksellä. Korpela toteaa tämän olevan hieman optimistista ajattelua, sillä näin virallinen standardi vaatii muun muassa käytäntöjen vakiinnuttamista ja varmuuden kaikkien piirteiden toimivuudesta. Ja ennen virallista standardia W3C:n pitäisi vielä julkaistava kielestä suositusehdotus (PR). (2014, 33.)

Kun puhutaan selainten HTML5-tuesta, tarkoitetaan lähinnä tukea niille HTML5-kielen määritellyille piirteille, jotka ovat lisäyksiä kielen aiempiin versioihin sekä selainten vanhaan käytäntöön (Korpela 2014, 90). Korpela kertoo jo IE 9 -version tukevan useita HTML5-kielen keskeisiä piirteitä, mutta kokonaisuudessaan tuki jää kuitenkin melko vaatimattomaksi. Valitettavasti iso, joskin vähenevä osa käyttäjistä ei pääse IE 8 -versiosta ylöspäin vaihtamatta konetta uudempaan, eikä välttämättä itse pysty

päivittämään selaintaan. (2014, 99.) Korpelan (2014, 90) mukaan nopea, mutta hyvin suurpiirteinen tapa tarkistaa selaimen HTML5-tuki on mennä verkkosivustolle html5test.com.

2.2 CSS3

Verkkosivujen muotoiluun käytettävä tyyliohjekieli CSS on kehittynyt uusimpaan muotoonsa, jota kutsutaan termillä CSS3. Vielä ei puhuta tyyliohjekieli-määritelmästä, koska CSS3-tyylikielestä ei ole tehty virallista määritelmää. CSS3 on terminä suhteellisen tuore, vaikka ensimmäinen luonnos siitä julkaistiin jo kesäkuussa 1999. Varsinainen CSS3-tyyliohjekielen kehittäminen alkoi jo CSS 2.1 -version julkistamisen aikoihin keväällä 2011. Kesäkuussa 2012 CSS työryhmä oli julkaissut CSS3-tyyliohjekielen jo yli 50 erilaista CSS-osaa. (Wikipedia 2014a, viitattu 3.5.2014.)

Web-konsortio (W3C), eli World Wide Web Consortium kehittää CSS3-tyyliohjekieltä ilman kilpailijoita, joten sen suhteen kehitystilanne on suhteellisen selkeä. (Korpela 2013, 14–15.) Korpela epäilee, ettei CSS3-tyyliohjekielestä julkaistaisi koskaan virallista määritelmää. Hän toteaa, että CSS3 on yleisnimitys kaikelle CSS 2.1 -version jälkeen tapahtuneelle CSS-tyyliohjekielen kehitykselle.

Itse optimisesti uskon, että CSS3-tyylikielestä tullaan vielä jossain vaiheessa julkistamaan virallinen määritelmä, koska suuri osa CSS3-tyylikielen osista on saavuttanut tason, joka voidaan määritellä lähes käytännön standardiksi.

Kuten Korpela kirjassaan mainitsi, CSS3 on vain yleisnimitys CSS-tyyliohjekielen kehitykselle. Ja vaikka CSS3:n osia onkin jo laajasti selaimissa toteutettu, se on edelleen kehitteillä. Se tuo CSS-tyyliohjekielen mukanaan muun muassa toiminnallisuuteen vaikuttavia ominaisuuksia. Monet CSS3-tyyliohjekielen ominaisuudet tukevat verkkosivujen ja -palveluiden yleisiä kehityssuuntia, joita ovat muun muassa laitteiden kirjon monipuolistuminen. (2013, 3 – 5.)

CSS3 rakentuu CSS 2.1 -version pohjalle ja se on säilyttänyt CSS-tyyliohjekielen yleisen muotorakenteen. Muutokset verrattuna CSS 2.1 -versioon ovat vähäisiä, sillä muutokset ovat pikemminkin määritelmien tarkentamista kuin muuttamista. Kuten edellä mainittiin, CSS3 on tuonut CSS-tyyliohjekielen mukanaan uusia arvoja vanhoille ominaisuuksille, mutta myös paljon aivan uusia ominaisuuksia sekä

rakenteita, jotka liittyvät laitteen ominaisuuksien mukauttamiseen. (Korpela 2013, 13 – 14.) Esimerkiksi mittayksiköitä on tullut lisää.

Haasteeksi myös CSS3-tyyliohjekielen kohdalla muodostuu se, että se toimii osittain vain jossain selaimissa, koska siitä ole tehty virallista tyyliohjekielimääritelmää. Etenkään Internet Explorerin (IE) vanhemmat versiot eivät tue sitä vielä kunnolla. Ongelmaksi muodostuu Korpelan mukaan se, että IE:n uusi versio voi vaatia uuden käyttöjärjestelmän asentamisen ja se taas vaatii usein uuden koneen. Esimerkiksi Windows XP -käyttöjärjestelmään ei saa enää IE 8 -versiota uudempaa selainversiota. (2013, 26.)

Kirjassaan Korpela toteaaakin, että CSS3-tyyliohjekieltä käytettäessä tulisi ottaa huomioon IE:n vanhemmat versiot, joissa CSS3-tuki voi olla hyvinkin puutteellinen. Korpela (2013, 16) kertoo sivuston caniuse.com kertovan selainkohtaisen tuen varsin havainnollisesti ja lisää, että niin opiskelu-, kokeilu- kuin tuotantovaiheessa on hyvä testata sivuston toimivuutta eri selaimilla.

Tätä ongelmaa on pyritty korjaamaan siten, että uusille CSS-ominaisuuksille ja ominaisuuksien arvoille on käytetty selainkohtaista etuliitettä. Esimerkiksi "border-radius" -ominaisuus toteutettiin Mozilla Firefox -selaimelle aluksi muodossa "-moz-border-radius". Tällaista käytäntöä on suositeltu selainten valmistajille niin kauan kun ominaisuus tai arvo ei ole saanut virallista hyväksyntää, eli saavuttanut tasoa CR. Tämä on aiheuttanut keskustelua puolesta ja vastaan. Toisaalta ajatellaan, että käytettäisiin vain standardin saaneita määritelmiä, mutta tämä taas supistaa huomattavasti uusien ominaisuuksien ja arvojen toimimista selaimissa. (Korpela 2013, 18.)

Korpela huomauttaa myös selaimien toimintatiloista, joilla voi vaikuttaa tyyliohjekielen toimivuuteen selaimessa. Nämä tilat ovat "standarditila" ja "oikkutila" (quirks mode, compatibility mode). Tyyliohjekielen toimivuuden kannalta kannattaa huomioida, että "oikkutilassa" selaimet eivät toimi CSS-määrittelyjen mukaan ja varsinkin IE jättää huomioimatta monet CSS3-tyyliohjekielen mukaiset ominaisuudet, jotka se kuitenkin tunnistaa "standarditilassa". Internet Explorerissa voi tarkistaa selaimen tilan F12-näppäimellä kohdasta Asiakirjan tila. (2013, 18.)

2.3 JavaScript

JavaScript on selaimessa toimiva, selainskriptien tekemiseen käytetty ohjelmointikieli. Selainskripti on ohjelmakoodi, jonka selain suorittaa HTML:n näyttämisen yhteydessä. JavaScriptiä voidaan käyttää monin tavoin, mutta yleisimmin sitä käytetään HTML-sivun toiminnallisuuden lisäämiseksi. Selainten JavaScript tuki on hyvin laaja, jonka vuoksi siitä on tullut käytännössä ainoa selainohjelmointikieli. (Korpela 2014, 55–56.)

JavaScript voidaan liittää HTML-sivuun monin tavoin, mutta Korpela suosittelee liittämään JavaScriptin ulkoisena tiedostona seuraavanlaisesti:

```
<script src="perusscriptit.js"></script>
```

Tällä tavoin ei synny ongelmaa siitä, miten erikoismerkit kuten "<" tai "&"-merkit kirjoitetaan, koska näillä merkeillä on HTML-kielessä oma erikoisasemansa. Ulkoiseen tiedostoon edellä mainitut merkit voidaan kirjoittaa sellaisinaan. Tämän tyyppisiä ongelmia ei Korpelan mukaan kuitenkaan juuri tapahtumamäärittelyssä synny, koska skripti on usein lyhyt ja yksinkertainen. Tapahtumamääritteiden käytökin voidaan kuitenkin korvata koodilla, joka kirjoitetaan ulkoiseen skriptitiedostoon. (2014, 57.)

2.4 jQuery

jQuery on pieni, nopea ja ominaisuuksiltaan rikas JavaScript-kirjasto. JQuery suorittaa HTML-dokumentin käsittelyt, tapahtumien käsittelyt sekä animoinnit paljon yksinkertaisemmin helppokäyttöisen dokumentaationsa (API) ansiosta. JQueryyn yhdistelmien monipuolisuus sekä laajennettavuus ovat muuttaneet tapaa, jolla miljoonat ihmiset JavaScriptiä kirjoittavat. (jQuery 2014, viitattu 17.9.2014.)

jQueryn tarkoitus on tehdä JavaScriptin kirjoittamisesta paljon helpompaa. jQuery on niputtanut JavaScriptin yleisimpiä, monta riviä vaativia komentoja metodeiksi, joita jQueryyn avulla voi kutsua vain yhdellä rivillä koodia. jQuery on yksinkertaistanut monia JavaScriptin monimutkaisia asioita, kuten Ajaxin kutsumista ja dokumenttioliomallin (DOM) käsittelyä. Näiden lisäksi jQuery sisältää HTML-kielen ja CSS-tyyliohjekielen käsittelyn, tehosteet ja animaatioita. (W3Schools 2014a, viitattu 17.9.2014.)

Vaikka maailmalla on lukuisa määrä erilaisia JavaScript-kirjastoja, jQuery näyttäisi olevan suosituin. Syytä suosioon lienee turha miettiä, sillä jQuery toimii kaikissa suurimmissa selaimissa, mukaan lukien Internet Explorerin versio 6. (W3Schools 2014a, viitattu 17.9.2014.) JQueryn rakenne eli syntaksi on tehty valitsemaan HTML-kielen elementtejä ja esittämään jotain toiminnallisuuksia elementissä. JQueryn perussyntaksi on seuraava:

```
$(selector).action();
```

\$-merkki osoittaa, että käytetään jQuery-kirjastoa. Selektori (selector) viittaa johonkin HTML-elementtiin ja action haluttuun toimintoon (Taulukko 2). (W3Schools 2014b, 28.9.2014.)

Taulukko 2. Esimerkkejä jQueryn syntaksista (W3Schools 2014b, viitattu 28.9.2014)

<code>\$("p").hide();</code>	Piilottaa kaikki p-elementit.
<code>\$(".test").hide();</code>	Piilottaa kaikki test-nimisen luokan (class) elementit.
<code>\$("#test").hide();</code>	Piilottaa elementin, jolla on id-määritelmä "test".

Elementtiin voidaan viitata myös this-määritteellä. Alla näkyvä esimerkki piilottaa siis aiemmin viitatus button-elementin. (W3Schools 2014c, 28.9.2014.) Tällöin esimerkkisyntaksi voisi olla seuraava:

```
$(document).ready(function(){  
    $("button").click(function(){  
        $(this).hide();  
    });  
});
```

3 ADOBE EDGE ANIMATE CC

Adobe Edge Animate CC on Adobe Systemsin kehittämä ohjelma, jolla voi tehdä päätelaitteisiin mu-
kautuvia vuorovaikutteisia animaatioita. Animate hyödyntää toiminnassaan HTML5-kieltä, JavaScriptia
(eritoten jQueryä) sekä CSS3-tyyliohjekieltä. Pääte CC tulee sanoista Creative Cloud, joka on Adoben
syyskuussa 2012 julkaisema pilvipalvelu. Ensimmäinen versio Adobe Edge Animatesta julkaistiin elo-
kuussa 2011, jolloin se oli vielä nimeltään Adobe Edge Preview. Uuden ohjelman kiinnostavuudesta
kielinee sekin, että 24 tunnin aikana julkaisusta ohjelmaa ladattiin yhteensä 50 000 kertaa. (Wikipedia
2014b, viitattu 11.7.2014.)

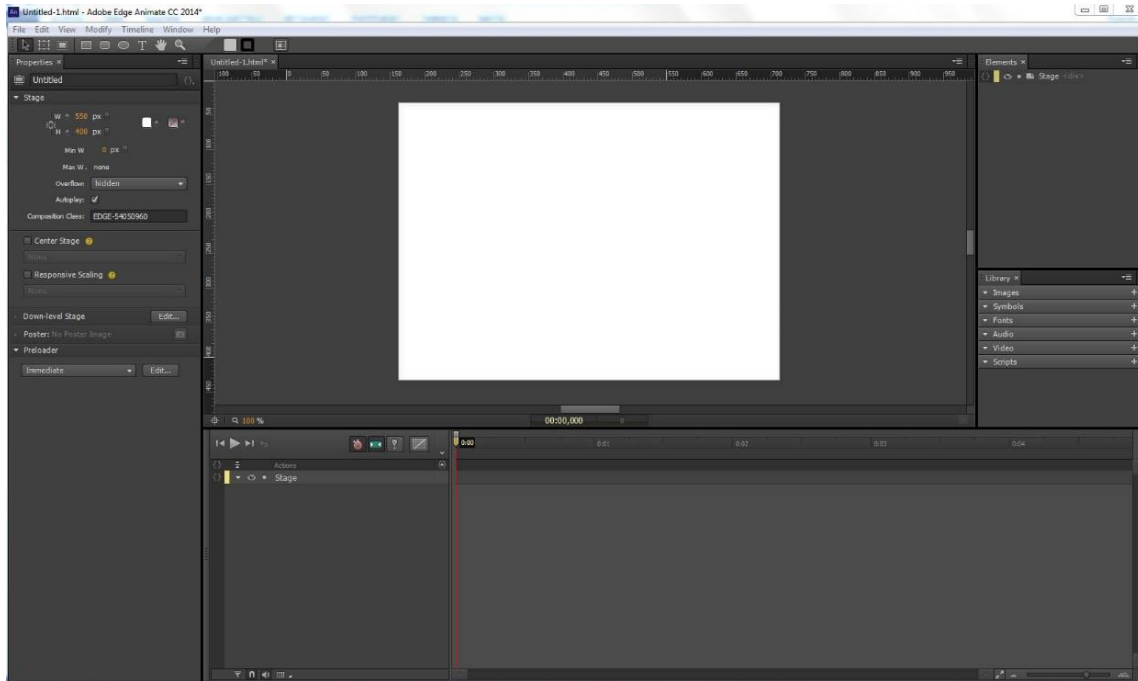
Lokakuussa 2011 Adobe ilmoitti, ettei tue Flashia enää mobiiliselaimissa ja samalla kertoi suunnitte-
vansa uusia tuotteita, jotka pohjautuisivat web-teknologioihin, kuten HTML5, CSS3, JavaScript sekä
jQuery. Syyskuussa 2012 Adobe Edge Preview sai nimekseen Adobe Edge Animate. Samaan aikaan
julkaistiin myös ohjelmat Edge Reflow, Edge Code ja Edge Inspect, jotka Animaten ohella kuuluvat
Adobe Edge -pakettiin, joka on kokonaisvaltainen web-sovelluskehitykseen tarkoitettu ohjelmistopa-
ketti. Tähän pakettiin kuuluvat myös apuohjelmat Edge Web Fonts, PhoneCap Build ja Adoben Typekit-
palvelu. (Wikipedia 2014b, viitattu 11.7.2014.)

Yhteensopivuus Internet Explorerin kanssa tuottaa web-kehittäjille päänvaivaa (Rohde 2013, 14–15).
Rohde kuitenkin lupaa Edge Animate CC -version toimivan kaikissa uusimmissa selaimissa, mukaan
lukien Chrome, Firefox, Safari sekä Internet Explorer 9. Tämä lisäksi se toimii kaikissa tableteissa sekä
älypuhelimissa, joihin on asennettu uusin selain. Äsken mainitut selaimet tukevat monia HTML5-kielen
ominaisuuksia, jonka lisäksi kyseisiin selaimiin on sisäänrakennettu automaattinen päivitys. (2013, 14–
15.) Seuraavaksi esitellään Edge Animaten käyttöliittymän ja toiminnan kannalta keskeisempiä osa-
alueita.

Stage

Kaikki toiminta tapahtuu Stagella (projektin ylin taso) ja tämä on alue, jolle animaatio rakennetaan.
Oletuksena Stagen pohjaväri on valkoinen (kuvio 1). Yksi Stagen parhaita ominaisuuksia on se, että

animaation voi katsella suoraan Stagella, koska se on HTML-ikkuna. Se toimii virtuaalisesti kuin selain, mutta on ohjelman sisään rakennettu. (Rohde 2013, 24.) Stagen asetuksista kerrotaan lisää luvussa 3.1.



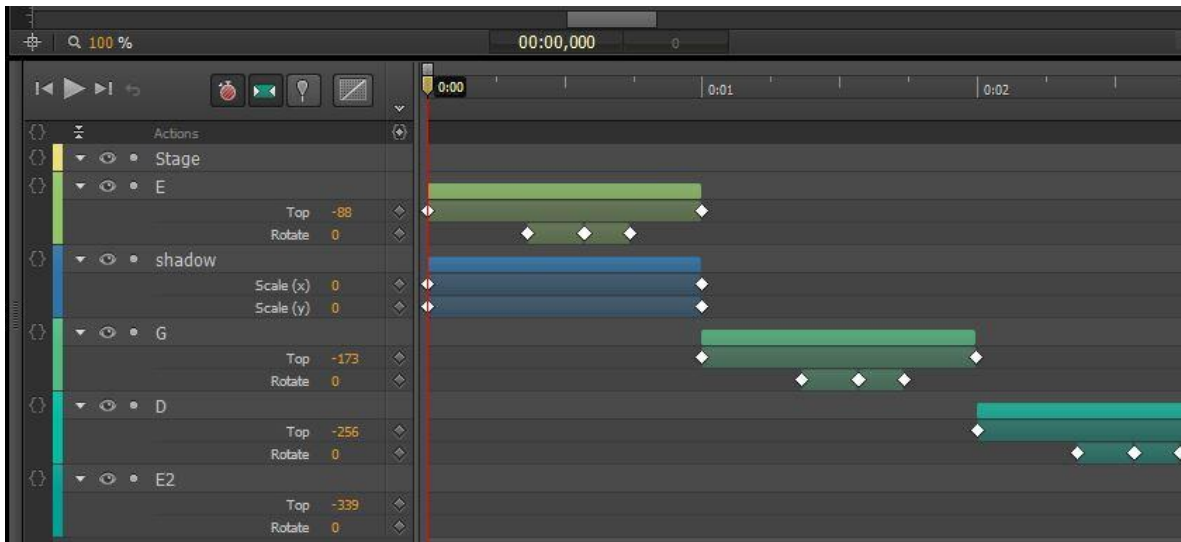
KUVIO 1. Projektin aloitusnäkö Adobe Edge Animatesta

Työkalurivi

Työkalurivi löytyy ohjelmaikkunan vasemmasta ylälaidasta (kuvio 1) ja se sisältää valintanuolen, muunnostyökalun, maskityökalun sekä piirtotyökalut - suorakulmion, pyöristetyn suorakulmion sekä ympyrän. Lisäksi työkaluriviltä löytyy tekstityökalu. (Rohde 2013, 26.)

Aikajana

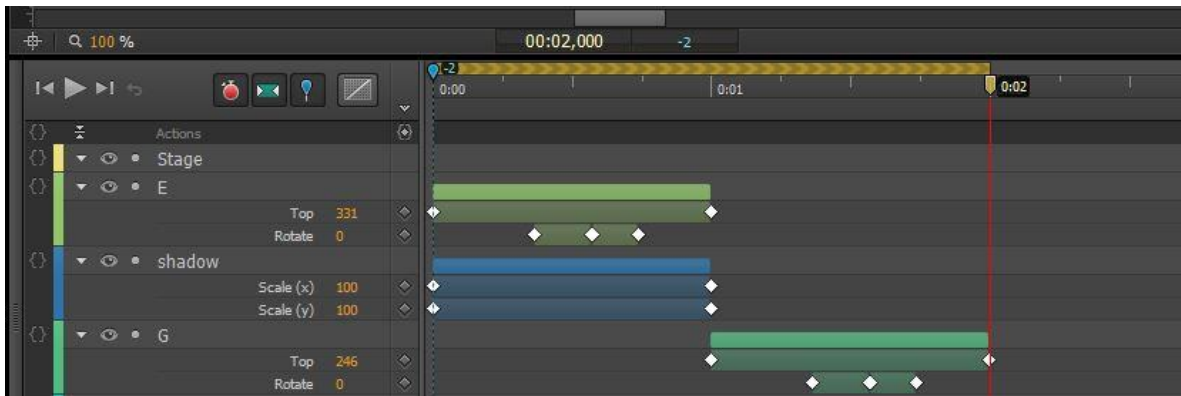
Stagen ja työkalurivin jälkeen käytetään varmasti eniten aikajanaa. Aikajana löytyy ohjelman alalaidasta (kuvio 2). Aikajanaa voi hallinnoida perinteisillä Play-, Rewind- ja Fast Forward -näppäimillä. Välilyönti-näppäin toimii pikanäppäimenä, jolla voi aloittaa tai lopettaa animaation toiston aikajanalla. (Rohde 2013, 56.)



KUVIO 2. Näkymä Animaten aikajanalta

Aikajan perusnäppäinten vierestä löytyvät animoinnin kannalta tärkeimmät aikajanäppäimet: Auto-Keyframe Mode, Auto-Transition Mode ja Toggle Pin (kuvio 2). Kukin näppäin muuttaa animoinnin luonnetta. Sekuntikellon näköisen symbolin takaa löytyvä Auto-Keyframe Mode tekee tapahtumapisteet (keyframe) aikajalle automaattisesti. Kun kyseinen symboli on aktivoituna ja liikuttaa elementtiä Stagella, tapahtumapisteet ilmestyvät aikajalle automaattisesti. Jotta kahden tapahtumapisteen välinen tapahtuma rekisteröityisi myös aikajalle, tulee aktivoida myös kirjekuoren näköisen kuvakkeen takaa löytyvä Auto-Transition Mode. Jos elementtejä halutaan liikuttaa Stagella, nämä kaksi näppäintä tulee olla kytkettynä pois päältä. (Rohde 2013, 59–60.)

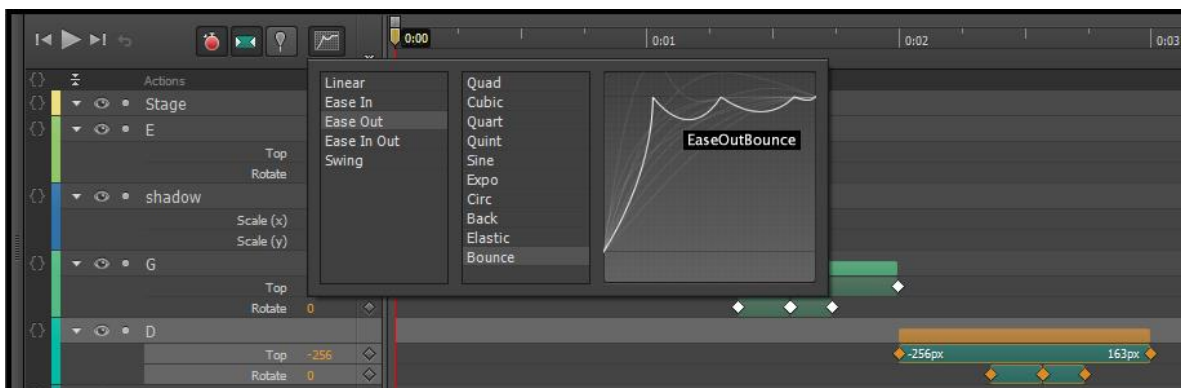
Nuppineula-symbolin takaa löytyvän Toggle Pin -työkalun avulla voi määritellä aikajalla, kuinka kauan animaatio kestää ja mihin suuntaan animaatio menee. (Rohde 2013, 61). Keskeistä Toggle Pin -työkalun käytössä on se, miten soittopään ja Toggle Pin -työkalun asettelee aikajalle. Esimerkiksi, kun halutaan animoida eteenpäin (kuvio 3), eli suuntaan, johon elementin halutaan liikkuvan tai muuttuvan Stagella, täytyy soittopää siirtää aikajalla siihen kohtaan, johon animaation halutaan kestävän ja tämän jälkeen liikuttaa tai muuttaa elementtiä Stagella halutulla tavalla. Syntyneen animaation huomaa aikajan keltaisesta palkista. Jos animaation halutaan olevan käänteinen, tulee siirtää vastavasti Toggle Pin -työkalua aikajalla haluttuun kohtaan, ja sen jälkeen liikuttaa tai muuttaa elementtiä halutulla tavalla. (Rohde 2013, 63–64.)



KUVIO 3. Toggle Pin -työkalun käyttäminen

Easing Tool

Easing Tool on reunimmainen näppäin aikajana-näppäinten vieressä. Tämän näppäimen takaa löytyy lukuisia erilaisia efektejä, jotka voi liittää napin painalluksella kahden keyframen väliseen animaatioon eli transaktioon. Esimerkiksi jos halutaan, että elementti pomppii Stagelle kuin pallo, voi valita Easy Out – Bounce (kuvio 4). Kunkin easing-asetuksen vieressä näkyy myös efektiin liittyvä grafiikka, joka kuvastaa efektin kulkua. (Rohde 2013, 61.)



KUVIO 4. Kahden tapahtumapisteen väliseen animaatioon, transaktioon, voi liittää easing-määrityksen

Properties (Asetukset)

Properties-paneeli sijaitsee käyttöliittymän vasemmalla reunalla ja sitä käytetään elementtien tylien muotoiluun ja lisäykseen. Elementit voivat olla mitä vain piirretystä suorakulmiosta kuvaan tai tekstiin.

Properties-paneelin asetukset muuttuvat riippuen valittuna olevasta elementistä. (Rohde 2013, 103–105.)

Elements-paneeli

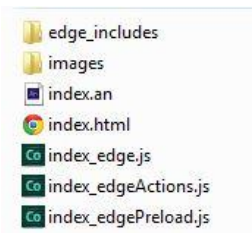
Elements-paneeli sijaitsee käyttöliittymän oikeassa yläreunassa. Elements-paneelissa voi tehdä lukuisia toimintoja, kuten valita ja lukita elementtejä, lisätä elementeille toiminnallisuuksia sekä ryhmittää elementtejä, tai laittaa niitä sisäkkäin. (Rohde 2013, 183.)

Library-paneeli

Library-paneeli sijaitsee Elements-paneelin alla. Paneeli helpottaa muun muassa kuvien, symbolien ja fonttien kanssa työskentelyä. Kun on Import-komennolla tuonut kuvia projektiin, ne ilmestyvät Library-paneeliin Images-osioon. (Rohde 2013, 193–194.) Library-paneelissa voi hallita myös symboleja. Symbolit mahdollistavat sisäkkäiset animaatiot, jotka toimivat omilla itsenäisillä aikajanoillaan ja sisältävät omat toiminnallisuutensa (Rohde 2013, 196).

Projektin kansiorakenne

Kun katsoo ohjelman automaattisesti luomaa kansiorakennetta, voi hämmentyä (kuvio 5). Animate huolehtii kuitenkin itse kansiorakenteen luomisesta. Kansiota löytyy an-päätteinen projektitiedosto sekä html-tiedosto projektin esikatselua varten. Edge_includes kansio sisältää kaksi JavaScript-tiedostoa, joista toinen on Edgen toimintaan liittyvä JavaScript-tiedosto ja toinen on jQuery-kirjasto, joka sisältää Animateen liittyvän API:n. Nämä tiedostot ovat samat jokaisessa projektissa. (Edge Docks 2014a, viitattu 29.8.2014.)



KUVIO 5. Näkymä Animate-projektin kansiorakenteesta

Kun projektiin tuodaan kuvia Import-komella, Animate luo automaattisesti projektikansioon images-kansion, josta löytyvät projektiin tuodut kuvat (kuvio 5). Kaikki kuvat, jotka projektiin tuodaan, tallentuvat tähän kansioon. Projektikansiosta löytyvä html-tiedosto on projektin pää-tiedosto. Tiedostoa on helppo laajentaa lisäämällä tiedoston head-elementtiin JavaScript-tiedostoja ja tyylitiedostoja. (Edge Docks 2014a, viitattu 29.8.2014.)

Kun projektin julkaisee, ohjelma luo projektikansioon publish-kansion. Projektin voi julkaista kolmeen eri muotoon, joista opinnäytetyössäni käytän Web-muotoista julkaisua. Tässä tapauksessa ohjelma luo publish-kansion alle web-kansion, jonka alta löytyy varsinainen html-tiedosto, images-kansio sekä nämä kolme edellä mainittua JavaScript-tiedostoa.

Projektikansiosta löytyy myös kolme erilaista JavaScript-tiedostoa (kuvio 5), jotka alkavat projektin nimellä - projektinnimi_edge.js, projektinnimi_edgeActions.js sekä projektinnimi_edgePreload.js. Näillä kaikilla tiedostoilla on oma tehtävänsä projektin lataamiseen ja esittämiseen liittyvissä vaiheissa. (Edge Docks 2014a, viitattu 29.8.2014.)

Ensimmäinen tiedosto, projektinnimi_edge.js ohjaa kaikkien elementtien ja symbolien rakennetta, keskinäisiä suhteita sekä yksityiskohtia. Se myös hallitsee ulkopuolelta ladatut kuvat sekä fontit. Tämä tiedosto on aika monimutkainen, joten sen käsittelyssä tulee olla hyvin varovainen. (Edge Docks 2014a, viitattu 29.8.2014.)

Seuraava tiedosto, projektinnimi_edgeActions.js, sisältää kaiken projektiin liittyvän toiminnallisuuden. Mikäli tuntee JavaScriptiä, tähän tiedostoon kannattaa tutustua lähemmin. Jos haluaa ymmärtää paremmin kuinka toiminnallisuuksia toteutetaan, voi luoda toiminnallisuutta ensin Animaten puolella ja sen jälkeen katsoa mitä muutoksia se sai aikaan kyseisessä JavaScript-tiedostossa. (Edge Docks 2014a, viitattu 29.8.2014.)

Kolmas, eli projektinnimi_edgePreload.js-tiedosto, hoitaa tiedoston lataamiseen liittyvät asiat, kun html-tiedosto on aukaistu selaimeen. Tähän tiedostoon ei tule tehdä muutoksia ja se on integroituna projektikansiossa olevaan html-tiedostoon. Animate ei tarvitse erillistä apuohjelmaa ohjelmakoodin suoritta-

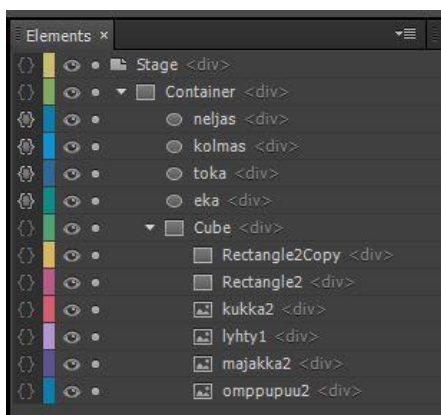
miseen, vaan se hyödyntää edellä mainittuja JavaScript-tiedostoja kollektiivisesti suorittaakseen ohjelman. Tästä toiminnasta käytetään nimitystä Adobe Edge Animate Runtime. (Edge Docks 2014a, viitattu 29.8.2014.)

Lyhykäisyydessään nämä kolme tiedostoa toimivat yhteen seuraavalla tavalla. Html-tiedosto avataan selaimella ja koodi suoritetaan. Html-tiedostoon sisällytetty projektinimi_edgePreload.js latautuu ja suorittaa ohjelmakoodin heti. Tämä ohjelmakoodi varmistaa, että tiedosto ladataan asianmukaisesti, jonka jälkeen Stagen rakenne, elementit ja symbolit ja niihin liittyvät toiminnallisuudet tulkitaan projektinimi_edge.js- ja projektinimi_edgeActions.js -tiedostoista. (Edge Docks 2014a, viitattu 29.8.2014.)

Joskus auttaa tai on välttämätöntä sisällyttää projektiin ulkopuolisia tyyli-tiedostoja tai kirjastoja. Edge Commons tai jQueryUI ovat hyviä esimerkkejä tällaisista kirjastoista. Animate ei itsessään hallinnoi näitä kirjastoja, joten ne täytyy tallentaa projektikansioon omaan kansioonsa. (Edge Docks 2014a, viitattu 29.8.2014.)

3.1 HTML ja Edge Animate

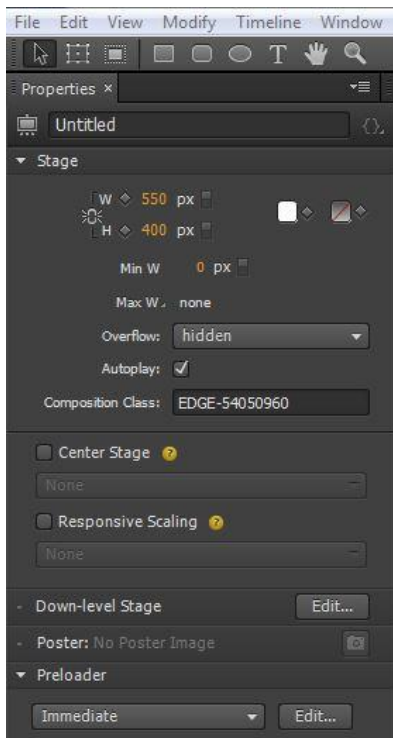
Edge Animate hyödyntää toiminnassaan HTML-kieltä. Tästä syystä kaikki elementit, jotka Stagelle piirretään tai tuodaan, ovat oletukseltaan div-elementtejä (kuvio 6) (Labrecque 2012, 103).



KUVIO 6. Näkymä Elements-paneelistä

Karkeimmillaan projekti on div-elementtejä div-elementtien sisällä. HTML-kielen merkitys ohjelman rakenteessa tulee esille, kun tutkitaan Properties-paneelin asetuksia. Kun Stage on valittuna, Properties-

paneelistä (kuvio 7) voi muuttaa muun muassa Stagen otsikon, joka on itseasiassa projektin luoman html-tiedoston title-elementti. Stagen asetuksista löytyy myös kyseisen tiedoston Composition ID, joka on tiedoston yksilöllinen tunniste. Ohjelma sisällyttää tämän tunnisteen automaattisesti ohjelman html-dokumenttiin. (Rohde 2013, 101–102.)



KUVIO 7. Näkymä Stagen Properties-paneelistä

Koska kaikki selaimet eivät vielä tue HTML5-kieltä, projektista voi ottaa varmuuskuvan (poster) (kuvio 7), jolloin käyttäjä näkee animaation sijasta still-kuvan jostain kohtaa animaatiota. Animaatiolle voi asettaa myös esilataus-toiminnon (Preloader), jota kannattaa käyttää silloin, kun projekti on raskas. (Rohde 2013, 102). Stagen asetuksissa on mukana myös CSS-määrittelyitä, joista kerrotaan tarkemmin luvussa 3.2.

Stagen asetuksiin verrattuna elementtien asetukset ovatkin jo hieman monimutkaisempia. Properties-paneelissa elementin ensimmäinen määritelmä on elementin id (kuvio 8). Elementistä riippuen elementin tyyppiä voi myös vaihtaa. (Rohde, 103–104.) Itse ohjelmaa käyttäessäni huomasin, että piirtotyöka-

luilla luodut elementit ovat div-elementtejä, eikä elementin tyyppiä voi vaihtaa. Kirjassaan Rohde kertoo, että projektiin tuotu kuvakin on oletuksena div-elementti, mutta sen voi vaihtaa img-elementiksi. Teksti-elementille voi sen sijaan vaihtaa lukuisia eri elementtityyppejä. (2013, 103–104.)



KUVIO 8. Div-elementti, jonka id-määritelmä Container

Varsinainen HTML5 liittyy Edge Animateen yksinkertaisuudessaan vain siten, että sen tuottama html-tiedosto on HTML5-muotoa. Kun avaa projektin html-tiedoston vaikkapa koodieditoriin, voi hämmästyä, kuinka yksinkertaista koodia tiedosto sisältää. Jo ensimmäisen rivin doctype-määrittelystä huomaa, että kyse on HTML5-tiedostosta. (kuvio 9). Ainoa projektiin liittyvä koodin pätkä lukee head-elementin sisällä Edge Runtime -kommenttien välissä, jossa kutsutaan tiedostonnimeä `index_edgePreload.js`-nimistä JavaScript-tiedostoa. Body-elementin sisällä on varsinainen projekti, eli div-elementti, jonka id-määritelmänä on Stage ja tämän class-määritelmänä on projektin automaattisesti luoma Composition ID. (Rohde 2013, 33.)

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
5   <meta http-equiv="X-UA-Compatible" content="IE=Edge"/>
6   <title>Untitled</title>
7   <!--Adobe Edge Runtime-->
8   <meta http-equiv="X-UA-Compatible" content="IE=Edge">
9   <script type="text/javascript" charset="utf-8" src="index_edgePreload.js"></script>
10  <style>
11    .edgeLoad-EDGE-10282447 { visibility:hidden; }
12  </style>
13  <!--Adobe Edge Runtime End-->
14
15 </head>
16 <body style="margin:0;padding:0;">
17   <div id="Stage" class="EDGE-10282447">
18   </div>
19 </body>
20 </html>
```

KUVIO 9. Animaten tuottama html-tiedosto

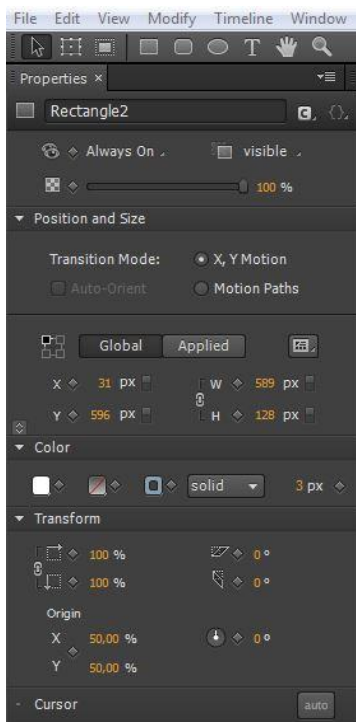
Työni kannalta HTML5-kielen merkitys tulee vastaan myös videobannerin toteutuksessa luvussa 5.4. Videobannerin teossa tulee ottaa huomioon HTML5-kielen tukemat videotiedostomuodot, MP4, WebM

sekä Ogg. Selainten tuessa on eroja ja edellä mainittuja tiedostomuotoja tukevat täysin vain selaimet Chrome ja Firefox. Esimerkiksi MP4-muotoa tukevat kaikki muut paitsi Opera. (W3Schools 2014d, viitattu 12.10.2014.)

3.2 CSS ja Edge Animate

Animatesta löytyy CSS3-tyyliohjekielen mukaisten määritysten lisäksi luonnollisesti myös jo standardisoidun CSS-tyyliohjekielen määrittäjiä. Animateissa CSS-tyyliohjekieli tulee vastaan elementtien Properties-paneelissa (kuvio 10). Properties-paneeli onkin tärkeässä roolissa animaation teon kannalta. Tässä kappaleessa esitellään opinnäytetyön kannalta tarpeelliset Properties-paneelin asetukset, joihin CSS-tyyliohjekieli liittyy.

Kaikille elementeille voi lisätä oman luokka-määrittelyn (class). Kun elementille on asetettu luokkamäärittely, pieni C-näppäin elementin id-määrittelyn oikealla puolella muuttuu valkoiseksi (kuvio 10). Luokkamäärittelyyn tarvitsee koskea yleensä vasta siinä vaiheessa, kun on tarve päästä käsiksi ulkopuolisiin JavaScript- ja css-tiedostoihin (Rohde 2013, 104).



KUVIO 10. Näkymä Properties-paneelista elementin ollessa valittuna

Stagen ja elementtien näkyvyyttä voidaan säätää muutamalla Properties-paneelin säätimellä (kuvio 10). Display-asetuksia säätämällä voidaan vaikuttaa elementin näkyvyyteen aikajanalla animaation aikana. Oletuksena display-arvo on "Always On", mutta elementti voidaan myös piilottaa "Off"-arvolla. Mikäli animaation aikana on säätänyt elementille joko "On"- tai "Off"-arvon, niin kytkettäessä päälle arvo "Always On", elementti näkyy koko ajan animaation ajan ja poistaa samalla aikajanalta aiemmat display-säädöt. (Labrecque 2012, 103.)

Display-säätimen vierestä löytyy overflow-säädin, jolla päätetään miten sisältö näytetään. Näytetäänkö se kokonaan (visible), vaikka sisältö menee yli reunojen, piilotetaanko se (hidden), kun määritelty alue loppuu vai ilmestyykö elementin viereen vierityspalkki, kun sisältö menee määritellyn alueen yli (auto). Scroll-asetuksella vierityspalkin saa näkyviin, vaikkei sisältö menisi määritellyn alueen yli. (Labrecque 2012, 103.)

Display- ja overflow-säädinten alta löytyy myös opacity-säädin, jonka avulla voi helposti säätää elementtien läpinäkyvyyttä. Opacity-määritelmä on yksi CSS3-tyyliohjekielen filter-määritelmän ominaisuuksista, joista kerrotaan lisää Properties-paneelin Filter-kohdassa (Rohde 2013, 130–138).

Position and Size sekä Transform

Properties-paneelistä löytyvät myös elementin koko- ja sijaintiasetukset (kuvio 10). Ohjelmaa tarkastellessa huomaa, että Animatessa elementtien muunnosasetukset on määritelty CSS3-tyyliohjekielen transform-määrittelyn mukaisesti.

CSS3-tyyliohjekielen kaksikulotteisen transform-määrittelyn avulla elementille voidaan määritellä sijainti (translate) top- ja left-arvoilla, sitä voidaan kääntää (rotate) tai vinottaa (skew). Määritteellä voidaan muuttaa myös elementin kokoa (scale) tai jopa yhdistää nämä kaikki edellä mainitut ominaisuudet (matrix). (W3Schools 2014e, viitattu 2.9.2014.) Animatessa näihin erikoisempiin transform-arvoihin pääsee käsiksi Properties-paneelin Transform-kohdassa, jossa voi muuttaa elementin kokoa (scale) sekä vinottaa (skew) tai kääntää (rotate) elementtiä. Elementin leveys ja korkeus säädetään W- ja H-säätimistä, jotka vastaavat CSS-tyyliohjekielen width- ja height-määrittelyjä.

Cursor

Properties-paneelistä elementille voi valita osoittimen (cursor), ja kaikkien osoitinvaihtoehtojen takaa löytyy jokin CSS-tyyliohjekielen cursor-arvo. Voi valita vaikka osoittavan sormen (pointer), joka viittaa linkkiin. Rohde (2013, 89) muistuttaa, että hyvät animaation tekijät antavat yleisölle paljon visuaalisia vinkkejä, koska näin yleisö ymmärtää helposti milloin ja missä elementissä on interaktiivisuutta.

Filters

Properties-paneelin kohta Filters on CSS3-tyyliohjekieltä vahvimillaan. Tässä kohdassa elementille voi määritellä käänteiset värit (invert), sävyn (hue), kontrastin (contrast), värikylläisyyden (saturate), mahdollisen seepia-sävyn (sepia), mustavalkoisuuden (grayscale), sumeuden (blur) sekä varjon (shadow). Filttäreillä saa monipuolisesti luotua erikoistehosteita, mutta Rohde (2013, 164) varoittaa, että filtlerin asetukset ovat vielä kokeiluasteella ja toimivat parhaiten selaimissa Chrome, Safari, iOS6 ja Blackberry 10.

3.3 JavaScript ja Edge Animate

Animaten yksi vahvimpia ominaisuuksia on sen kyvykkyys lisätä animaatioon toiminnallisuutta (actions), laukaisimia (triggers) ja kylttejä (labels). Juuri nämä työkalut tekevät animaatiosta interaktiivisen. (Rohde 2013, 77.) Toiminnallisuutta voidaan lisätä niin yksittäiseen elementtiin, koko Stagelle tai vain tiettyihin kohtiin aikajanalle. Kuten jo luvussa 3 kerrottiin, JavaScript, eritoten jQuery ovat Animaten toiminnan kannalta hyvin oleellisia. Kaikki koodi Actions-paneelissa on JavaScript-kieltä ja toiminnallisuuksissa hyödynnetään jQueryä sekä Adobe Edge Animate Runtime API:a. (Labrecque 2012, 194.)

Triggers

Laukaisin (trigger) on aikajanalla sijaitseva komento animaatiolle ja se kertoo kuinka animaation pitää juuri tietyn laukaisimen kohdalla käyttäytyä. Tyypillisin trigger-komento on käskää animaatiota toistumaan, kun animaatio on loppunut (looping). (Rohde 2013, 78.)

Labels

Kylttejä (labels) voi lisätä aikajanalle, käytettiinpä sitten laukaisimia (triggers) tai elementtien toiminnallisuuksia (actions). Esimerkiksi tietyn laukaisimen kohdalla voi käskää animaatiota hyppäämään tietyn kyltin kohdalle. Rohden (2013, 81) mukaan kylttejä on helppo käyttää, koska niitä voi liikuttaa aikajalla edestakaisin haluamaansa kohtaan, eikä koodiin tarvitse koskea lainkaan.

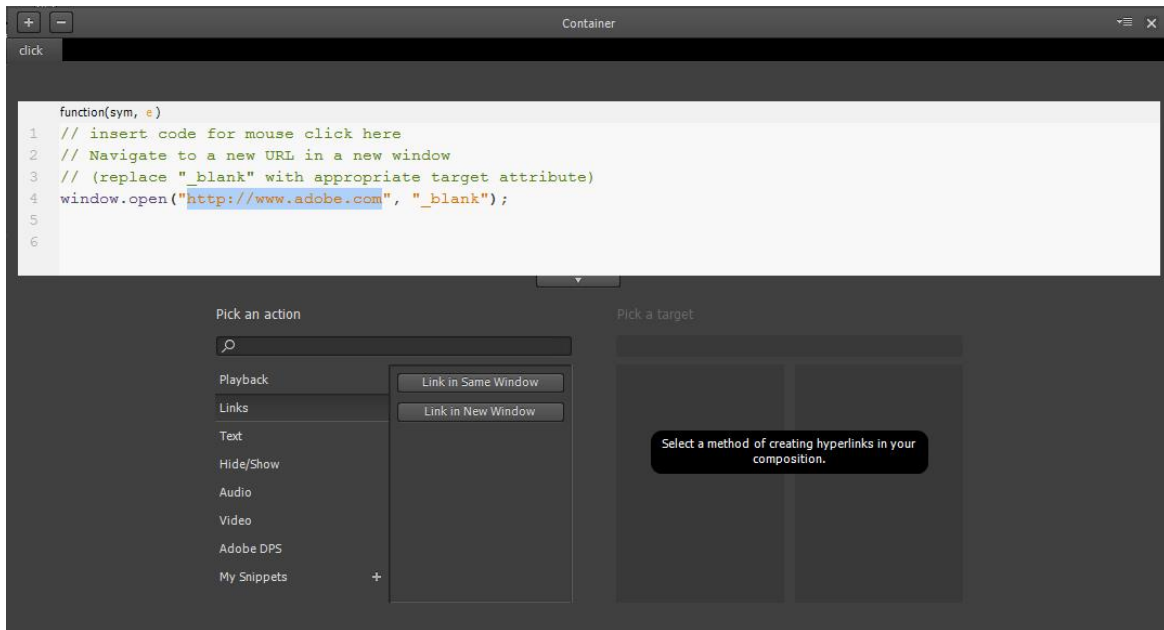
Actions

Actions-ikkunasta löytyy lukuisa määrä erilaisia vuorovaikutteisia toimintoja, joita ovat muun muassa hiiren klikkaus, hiiren tuplaklikkaus, hiiren liikkeet, pyyhkäisy (swipe) ja kosketus (touch) (kuvio 11). Mobiilikäyttäjiä ajatellen pyyhkäisytoiminnallisuutta ja kosketustoiminnallisuutta voi käyttää samalla tavoin kuin klikkaustoimintoa. Rohde (2013, 84) jopa suosittelee käyttämään klikkaustoimintoa, mikäli animaation on tarkoitus toimia sekä mobiilissa että pöytäkoneella.



KUVIO 11. Elementin actions-ikkunan toiminnan laukaisijat

Kun toiminnallisuuden laukaisija on valittu, valitaan suoritettava toiminto. Toiminto voi olla esimerkiksi elementin klikkauksesta avautuva uusi välilehti johonkin tiettyyn URL-osoitteeseen (kuvio 12). (Rohde 2013, 85–86.)



KUVIO 12. Näkymä Actions-paneelistä

Stagella on normaalien actions-komentojen lisäksi omat ainutlaatuiset toiminnallisuutensa, jotka osaltaan vaikuttavat animaation kulkuun. Näistä toiminnallisuuksista mainittakoon esimerkiksi Creation Complete, joka suorittaa toiminnallisuuden heti, kun Stage on ladattu sekä CompositionReady, joka suorittaa toiminnallisuuden heti, kun koko projekti on ladattu ja valmis toimintaan. Lisätietoja näistä Stagen omista ja Adobe Edge Animaten yleisistä toiminnallisuuksista saa Adobe Edge Animate Runtime API-dokumentaatiosta. (Rohde 2013, 184–185.) Tuorein dokumentaatio löytyy kirjoittamalla Googleen hakusanaksi Adobe Edge Animate Runtime API.

Animateen voi liittää myös ulkopuolisia JavaScript-tiedostoja ja -kirjastoja. Yksi esimerkki tällaisesta ulkopuolisesta kirjastosta on vapaaseen lähdekoodiin perustuva Edge Commons, josta löytyy monta Animaten työkulkua parantavaa ja helpottavaa osaa. Edge Commons sisältää monta erilaista JavaScript-tiedostoa. Sivusto kuitenkin kehottaa uusia käyttäjiä käyttämään tiedostoa, jossa on kaikki Edge Commonsin sisältämät lisätoiminnallisuudet. (EdgeDocks 2014b, viitattu 3.9.2014.)

Toinen tehokas Animaten laajentaja on Edgehero-niminen JavaScript-kirjasto, joka osaltaan helpottaa työkulkua Animate-projekteja tehdessä. Edgehero tukee muun muassa kolmiulotteisuutta, HTML5-videoita sekä Youtube-videoiden upottamista. (Edgehero, viitattu 11.10.2014.)

Animaten toiminnallisuuksia voi laajentaa myös animointiin erikoistuneen JavaScript-kirjasto Green-Sock Animation Platformin (GSAP) avulla, joka on paketti työkaluja ohjelmoituun animointiin. (Green-sock 2014, viitattu 3.9.2014). Näistä ulkopuolisten JavaScript-kirjastojen sisältämistä lisätoiminnallisuuksista ja käytöstä kerrotaan enemmän luvussa 5.

Koodi-paneeli

Code- eli koodipaneeli (kuvio 13) löytyy ohjelmasta Window-Code. Tästä paneeli-ikkunasta näkee kaikille elementeille (actions) ja aikajanelle (triggers) määritellyt toiminnallisuudet. Tämän ikkunan kautta toiminnallisuuksia on myös helppo lisätä tai poistaa. Mikäli omaa ohjelmointitaitoja, voi tämän paneelin kautta myös muokata toiminnallisuuksia. Tämä onnistuu paneelin oikeassa laidassa olevasta "Full Code" -napista, joka näyttää projektinnimi_edgeActions-tiedoston muokattavassa muodossa. (Lab-recque 2012, 206–207.)



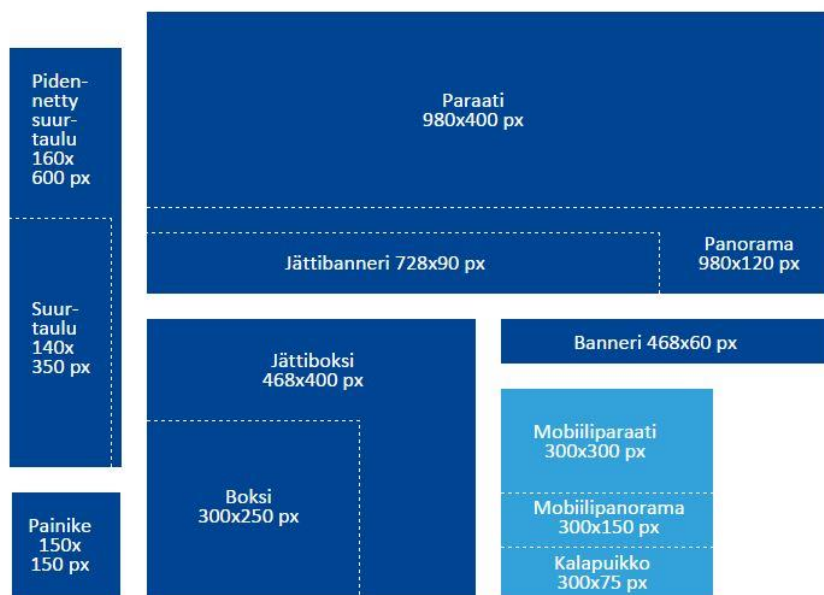
KUVIO 13. Näkymä Code-paneelistä

4 DISPLAY- ELI BANNERIMAINONTA

Display- eli bannerimainonta yksi vanhimpia ja tunnetuimpia verkkomainonnan muotoja. Bannerit ovat erikokoisia, suorakaiteen muotoisia verkkosivustolle sijoitettuja mainoksia, jotka yleensä sisältävät lyhyen animaation. (KWD Digital 2014a, viitattu 5.9.2014.)

Suomessa verkkomainonnan asiantuntijajärjestönä toimii IAB Finland, jonka tehtävänä on edistää digitaalisen mainonnan kasvua ja kehitystä. Järjestön jäsenet koostuvat verkkomediaista, mainostoimistoista, interaktiivisista toimistoista, mediatoimistoista, verkkomedioiden myyntiyhtiöistä, tutkimusyryksistä, mainostajista ja teknologiaa kehittäivistä yrityksistä. IAB Finland on IAB Europen jäsen. IAB Europe on digitaalisen mainonnan kattojärjestö, jonka jäsenistöön kuuluu yli 5 500 eurooppalaista jäsenyristä. (IAB Finland 2014a, viitattu 26.10.2014.)

IAB on määritellyt verkkomainonnan standardikoot (kuvio 14), joita display-mainonnassa yleisimmin käytetään (IAB Finland 2014b, viitattu 5.9.2014). Bannerit toteutetaan yleensä standardikokoisina hyvän ja laajan yhteensopivuuden takaamiseksi (Järvilehto 2012, 97).



KUVIO 14. IAB:n määrittelemät display-mainoskoot

IAB on määritellyt muutamille display-muodoille myös suosituksia, jotta verkkomainonnan käytännöt olisivat mahdollisimman yhtenäiset. Esimerkiksi paraati-mainoskoolle on määritelty suositelluksi kilotavurajaksi 80 kt. Ja jos mainoksessa on videota, se ei saa latautua automaattisesti, vaan vasta silloin kun käyttäjä vie hiiren mainoksen päälle. (IAB Finland 2014c, viitattu 5.9.2014.)

Display-mainontaa voidaan hinnoitella useilla eri tavoilla - mainosnäyttöjen mukaan sekä aika-, klikkaus- tai toimintaperusteisesti. Mainosnäyttöjen mukainen hinnoittelu (cost per mille, CPM) tarkoittaa kuinka monta kertaa mainosta näytetään, esimerkiksi tuhat kertaa. Aikaperusteisessa hinnoittelussa mainosta näytetään sivustolla tietyn aikaa, esimerkiksi yhden kuukauden ajan. Klikkausperusteinen hinnoittelu (cost per click, CPC) perustuu klikkausten määrään. Tässä tapauksessa mainospaikka ei maksa mitään, mutta joka klikkauksella on tietty hinta. Toimintaperusteisessa hinnoittelussa hinnoittelu (cost per action, CPA) perustuu vain mainoksen kautta tehtyihin ostoksiin, esimerkiksi osto verkkokaupasta. (KWD Digital 2014b, viitattu 13.9.2014.)

Viime vuosien ajan display-mainonta on ollut vahvassa kasvussa. IAB:n huhtikuisen lehdistötiedotteen mukaan display-mainonta oli kasvanut ensimmäisellä vuosineljänneksellä lähes 20 %. Yleisesti ottaen verkkomainonnan osuus kaikesta mainonnasta on jo lähes 24 %. Display-mainonnan osuus verkkomainonnasta on reilu 40 %. (IAB Finland 2014d 2014, viitattu 16.9.2014.)

4.1 Display-mainonta Kaleva Oy:ssä

Vuonna 2013 Kalevan verkkosivusto tavoitti keskimäärin 240 000 eri kävijää viikossa. Display-mainoksia myytiin verkkosivustolle yhteensä noin 1600, joista valmistettavia oli noin 25 % (Peltokorpi 17.9.2014, sähköpostiviesti). Display-mainonnan osuus Kalevan tuloksesta on 5 prosenttia ja osuus kasvaa vuosittain noin 30 prosenttia (Haataja 11.9.2014, haastattelu).

Vuoden 2014 viikolla 43 Kalevan verkkosivusto oli seitsemänneksi suosituin sanomalehtien verkkopalvelu. Tuolloin Kalevan verkkosivustolla vieraili lähes 329 000 eri kävijää ja sivulatauksia oli hieman yli 3,3 miljoonaa. Viikolla 44 sivulatauksia oli hieman yli 4 miljoonaa. (TNS Metrix 2014, viitattu 7.11.2014.) Loka-marraskuun välisenä aikana istunnoissa käytettiin Chromea 30 %, Firefoxia 24 %, Internet Ex-

ploreria 22 % ja Safaria 12 %. Internet Explorerin osalta käyttö jakaantui sitten, että versiota 11 käytettiin 56 %, 10-versiota ja 9-versiota lähes saman verran 18–19 %. IE 8 -versiota käytettiin enää vain vajaa 5 %. (Haataja 6.11.2014, sähköpostiviesti.)

Kalevan verkkosivustolta voi ostaa näkyvyyttä näyttöperusteisesti eri mainospaikoille läpi sivuston tai kohdistetusti etusivulle, uutisiosioihin esimerkiksi Oulu, Pohjois-Suomi, Urheilu, Talous ja teemasivuille esimerkiksi Koti, Moottori ja Kiekko-Kaleva (DigiKaleva 2014a, viitattu 17.9.2014). Tämän lisäksi mainoksia voidaan kohdentaa muun muassa käyttöjärjestelmän, domainin, maantieteellisen sijainnin, yhteystavan, selaimen ja/tai selaimen kielen mukaan. (DigiKaleva 2014b, viitattu 17.9.2014.)

Kalevan verkkosivuston suosituimmat display-mainoskoot ovat sivuston yläosaan sijoitetut paraati, panoraama ja boksi. Yhtenä kehityssuuntana on, että asiakkaat haluavat huomiota herättävämpiä mainoskampanjoita. Tämä tarkoittaa muun muassa suurempia mainospintoja, joilla kampanjalle pyritään saamaan suurempi huomioarvo. (Haataja 11.9.2014, haastattelu.)

Aineiston siirtoon Kaleva käyttää verkkomainonnan hallintaan tarkoitettua, selaimessa toimivaa Cxense-ohjelmistoa, johon Kalevan verkkosivuille tuleva display-mainosaineisto syötetään. Jokaiselle Kalevan verkkosivustolla näytettävälle bannerille tehdään oma varaus, jossa määritellään muun muassa bannerin koko sekä bannerin näyttömääriin ja kohdennukseen liittyvät asetukset.

Ohjelmistolla on kolme vaihtoehtoa HTML5-aineiston syöttämistä varten. Aineiston voi lisätä suoraan halutulle verkkosivustolle, syöttää kaikki html-tiedostot suoraan Cxense-ohjelmistoon tai isännöidä HTML-tiedostoja itse. (Emediate Classroom, viitattu 16.10.2014.)

Tällä hetkellä Kaleva isännöi HTML5-tiedostoja itse, pilvessä toimivan tiedostonhallintaohjelman kautta. Aineisto siirretään kyseiseen tiedostonhallintaohjelmaan, jossa aineistolle tehdään julkaisuun tarvittavat toimenpiteet. Ohjelmasta saadaan linkki aineistoon, joka syötetään Cxense-ohjelmiston varaukseen.

4.2 Display-mainonnan tulevaisuus Kaleva Oy:n näkökulmasta

Jo 40 % Kalevan lukijoista on mobiilikäyttäjiä. Kaleva onkin ryhtynyt muutostöihin koskien Kalevan verkkosivustoa, josta tehdään parasta aikaa responsiivista, eli päätelaitteeseen mukautuvaa. Muutoksen on tarkoitus valmistua vuoden 2014 loppuun mennessä. Responsiivinen verkkosivusto tuo mainostajille paremman näkyvyyden, koska uudistuneella verkkosivustolla mainospaikkoja tullaan jonkin verran vähentämään. Kalevan verkkosivuston responsiivisuuden myötä standardin mukaiset mainospaikat eivät voi kuitenkaan muuttua responsiivisiksi. Sen sijaan osa mainospaikoista muuttuu adaptiivisiksi, mikä tarkoittaa, että bannerin koko muuttuu täysin näytön tiettyjen leveysmääritelmien mukaisesti. (Haataja 11.9.2014, haastattelu.)

Adaptiivisessa suunnittelussa käytetään staattisia asetelmia, jotka perustuvat tiettyihin pysäytyspisteisiin (breakpoints), kun taas responsiivisessa suunnittelussa asetelma muuttuu koko ajan sitä mukaan kun selaimen ikkunan koko muuttuu. Eli adaptiivisessa suunnittelussa tunnistetaan laite ja kutsutaan sopivan kokoista asetelmaa. (Williams 2013, viitattu 5.10.2014.)

Otetaan esimerkiksi kaleva.fi-sivuston ylämainospaikka. Näytön leveyden ollessa yli 980 pikseliä, ylämainospaikalla näytetään joko paraati- tai panoraama-kokoinen banneri. Kun näytön leveys on alle 980 pikseliä, eli esimerkiksi tabletin pystynäyttö, niin ylämainospaikalla näytetään jättiboksin kokoinen banneri. Kun näytön leveys on alle 768 pikseliä, niin ylämainospaikalla näytetään boksi-kokoinen banneri.

Tällä hetkellä Kaleva myy display-mainoksia näyttöperusteisesti. Suunta tulee Haatajan (11.9.2014, haastattelu) mukaan olemaan tulevaisuudessa varmasti enemmän klikkiperusteinen, kun siirrytään rikkaampaan sisältöön. Tässä yhteydessä on hyvä mainita termi ”Rich media”, joka on nimitys mainosmuodoille, joissa käyttäjälle näytetään staattisen tai animoidun kuvan sijasta elävää kuvaa ja ääntä, usein sisältäen interaktiivisia elementtejä (Mediareaktori 2014, viitattu 16.9.2014).

Kalevan pyrkimyksenä on saada tehtyä toimivampia display-mainoksia, jotta mainostajat huomaavat, että verkkomainonta on tehokas tapa mainostaa. Ideaalitalanne Haatajan mielestä ovat monipuoliset mediaratkaisut, joissa mainostaja pyrkisi hyödyntämään mainonnassaan niin printti- kuin display-mainontaa. (Haataja 11.9.2014, haastattelu.)

5 BANNERIPOHJIENTOTEUTUS

Opinnäytetyön kehitystehtävänä oli suunnitella ja toteuttaa Adobe Edge Animatella neljä erilaista banneripohjaa (template) Kaleva Oy:n markkinoinnillisiin tarpeisiin. Kukin näistä bannereista sisältää erilaisia ominaisuuksia. Tärkeimpiä vaatimuksia banneripohjille olivat toimivuus eri selaimissa sekä päätelaitteissa.

Toteutuksessa tuli ottaa huomioon banneripohjien yleinen työnkulku - valmistuksesta bannerin julkaisuun ja aineiston siirtämiseen Kalevan verkkosivustolle. Suunnittelussa ja toteutuksessa tuli ottaa huomioon myös tuotannon näkökulma, eli banneripohjien tuli olla helppoja muokata ilman ohjelmointitaitoja. Toteutukseltaan bannereiden tuli vastata tulevaisuuden tarpeita, joita ovat erilaiset rich media - toteutukset, kuten kolmiulotteisuus, interaktiivisuus ja videosisältö.

Selaintuen testaus

Testaukseen käytettiin suhteellisen vähän aikaa, sillä muutoin kehitystehtävät ja työ olisivat paisuneet liian isoiksi. Luotettiin siihen olettamukseen, että Animatella tehdyt bannerit toimivat uusimmilla selaimilla. Banneripohjat käytettiin kuitenkin Kalevan testisivustolla, jossa pystyi näkemään ja testaamaan banneripohjien toimivuuden ja toiminnallisuuden niin pöytäkone-ympäristössä kuin mobiililaitteissa. Bannereiden toimivuus testattiin yleisimpien selainten uusimmilla versioilla mukaan lukien Chrome, Firefox, IE11 sekä Safari.

Mukautuminen päätelaitteisiin

Kalevan verkkosivuston uudistamisen vuoksi yhdeksi tärkeäksi ominaisuudeksi nousi banneripohjien adaptiivisuus, josta kerrottiin luvussa 4. Adaptiivisuutta käytettiin esimerkin vuoksi työn yhdessä banneripohjassa, jossa adaptiivisuuden katsottiin olevan tarkoituksenmukaista. Adaptiivisuuden toteuttamisesta kerrotaan tarkemmalla tasolla luvussa 5.2.

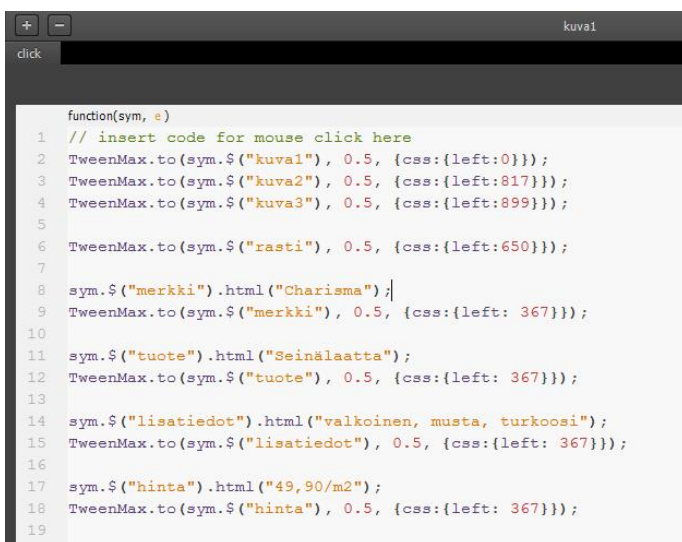
5.1 Haitaribanneri

Tämän banneripohjan (liite 1) idea on varsin yksinkertainen. Siinä on kolme kuvaa, joita kutakin klikkaamalla saa muut kuvat siirtymään ja tekstin liukumaan näkyviin. Tämän banneripohjan tekee erikoiseksi se, että siinä ei käytetä lainkaan aikajanaa. Aikajanan sijaan käytettiin Greensock-nimistä JavaScript-kirjastoa. Ulkoisen JavaScript-tiedoston voi liittää Edge Animateen näppärästi Library-paneelin Scripts-osiosta, joka on paneelin alimmaisena. Tiedoston voi liittää joko olemassa olevasta kansiossa tai URL-osoitteen perusteella. Projektiin liitettiin Greensockin internetsivustolta saatu CDN-linkki TweenMax-nimiseen JavaScript-tiedostoon (kuvio 15).



KUVIO 15. Edge Animateen Library-paneelin kautta liitetty JavaScript-tiedosto

Bannerin toiminnallisuus perustuu hiiren klikkaukseen. Toiminnallisuuden aikana kutsutaan TweenMax-tiedoston funktiota, joka määrittelee, kauanko määritellyn elementin toiminnallisuus kestää ja kuinka elementtiä halutaan muuttaa. Tässä banneripohjassa hyödynnettiin funktiota vain elementtien sulavaan liikkumiseen elementin left-arvoa muuttamalla (kuvio 16). Kyseistä JavaScript-kirjastoa haettiin kokeilla, koska sen avulla pienillä koodin muutoksella saa tehtyä toiminnallisuuden, joka vaatisi Animaten aikajanalla tehtynä monta työvaihetta.



KUVIO 16. Näkymä ensimmäisen kuvan click-toiminnallisuudesta

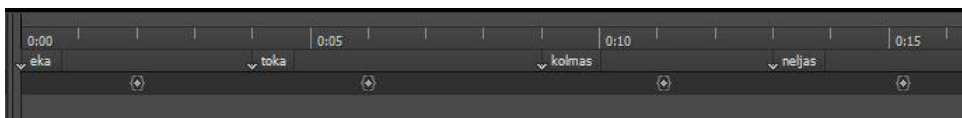
Jokaiseen kuvaan toteutettiin klikkaus-kehote, jotta käyttäjä huomaa bannerissa olevan toiminnallisuutta. Kutakin kuvaa klikkaamalla saadaan muut kuvat siirtymään ja teksti liukumaan näkyviin. Kuvien tuli palautua myös paikoilleen. Toiminnallisuuden osalta tuli ottaa huomioon myös bannerin toimivuus mobiililaitteissa. Tätä varten banneripohjaan toteutettiin kaksi toiminnallisuutta. Hiiritoiminto, mouseleave, jolloin kuva palautuu takaisin paikalleen, kun hiiri viedään pois bannerin päältä. Tämä ei luonnollisestikaan onnistu mobiililaitteille, joten mobiilikäyttäjiä ajatellen jokaiseen kuvaan toteutettiin myös sulje-nappi (rasti), jota painamalla kuva palaa takaisin alkutilanteeseen.

Usein bannerista pääsee ulkopuoliselle verkkosivustolle ja näin olleen bannerin alaosa on varattu mainostajan logolle, jota klikkaamalla pääsee esimerkiksi mainostajan verkkosivustolle. Alapalkkiin voi halutessaan sijoittaa myös muita linkkejä.

5.2 Slider

Tämän banneripohjan (liite 2) ideana on neljä erilaista näkymää, joita voi hallita eteen- ja taaksepäin-nuolilla. Tämä banneripohja toteutettiin adaptiivisena, koska tässä banneripohjassa sen toteuttaminen oli tarkoituksenmukaista. Banneripohjien ulkoasu toteutettiin Kalevan graafisen suunnittelijan tekemien suunnitelmien mukaan. Bannerin toiminnallisuuksien tekeminen ei vaatinut toimiakseen ulkopuolisia JavaScript-tiedostoja, vaan kaikki toiminnallisuus rakennettiin aikajanaa ja siihen liitettäviä toiminnallisuuksia, kuten kylttejä ja laukaisimia hyödyntäen. Adaptiivisuuden toteuttamisessa sen sijaan käytettiin Edge Commons -nimistä JavaScript-kirjastoa.

Bannerissa tekstit ja kuvat vaihtuvat nuolinäppäimiä painettaessa. Näkymien vaihtuvat tekstit ja kuvat animoitiin aikajanelle lineaariseen järjestykseen. Banneria ohjataan eteen- ja taaksepäin-nuolilla, joten jokaisen näkymän alkuun täytyi asettaa kyltit (labels). Kylttien lisäksi aikajanelle asetettiin pysäytyslaukaisimet niihin kohtiin, joihin kunkin näkymän animaatio päättyi (kuvio 17).



KUVIO 17. Aikajanelle määritellyt kyltit (labels) ja laukaisimet (triggers)

Kullekin näkymälle tehtiin omat eteen- ja taaksepäin-nuolet, joiden näkyvyyttä muutettiin display-säätimellä. Esimerkiksi ensimmäisen näkymän eteenpäin-nuolelle annettiin klikkaustoiminnolla käsky soittaa animaatio toisen kyltin kohdalta (kuvio 18) ja taaksepäin-nuolelle käskyn soittaa animaation neljännen kyltin kohdalta.



KUVIO 18. Ensimmäisen näkymän eteenpäin-nuolen klikkaus soittaa animaation toisen kyltin kohdasta

Adaptiivisuutta varten bannerista tehtiin kolme erikokoista asetelmaa (layout). Tässä tapauksessa suurin koko oli paraati, eli 980 x 400 pikseliä, keskimäinen oli jättiboksi 468 x 400 pikseliä ja pienin näytettävä koko oli boksi eli 300 x 250 pikseliä. Bannerin kahdesta pienemmästä bannerikoosta tehtiin toiminnallisuksiltaan yksinkertaisemmat. Kaikista kolmesta animoidusta asetelmasta tehtiin symbolit. Näille symboleille täytyi määritellä maksimi- ja minimileveysasetukset, jotta ne toimisivat oikein. Symbolit nimettiin siten, että paraati-kokoinen symboli oli nimeltään layout980, jättiboksi layout768 ja boksi layout300.

Animatella saa tehtyä bannerista responsiivisen muutamia asetuksia muuttamalla, mutta se, että bannerin koko ja sisältö muuttuisi tiettyjen leveysmääritelmien mukaan, eli adaptiivisesti, vaatii ulkopuolisen JavaScript-kirjaston Animaten tueksi. Adaptiivisuuden toteuttamiseen käytettiin Edge Commons -nimistä JavaScript-kirjastoa ja sieltä funktiota nimeltään Adaptive Layouts. Kirjaston käyttöönottaminen vaatii vain muutaman rivin ohjelmointia, ja siihen ohje löytyi suoraan Edge Commonsin sivuilta.

Tätä JavaScript-kirjastoa ei ladattu aiempien banneripohjien tapaan Libraryn Scripts-osioon, vaan siitä tehtiin Stagelle toiminnallisuus nimeltä CreationComplete, joka tarkoittaa, että koodi suoritetaan, kun Stage on luotu. Funktio osaa käsitellä layout-etuliitteen ja näin hakea näytölle oikean kokoisen bannerin. (kuvio 19).



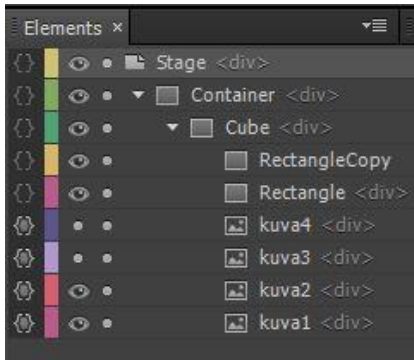
KUVIO 19. Adaptive Layouts-funktiota kutsutaan, kun Stage on luotu

5.3 3D-kuutiokaruselli

Tämä banneripohja (liite 3) on kolmiulotteinen kuutio. Kuutio sisältää neljä kuvaa ja kuutio kääntyy 360 astetta X-akselinsa ympäri, näyttäen yhtä sivua tietyn ajan. Lisäksi kuutioon haluttiin linkkitoiminnallisuus, eli jokaisesta kuution sivusta pääsisi niihin määritellylle internet-sivustolle. Näin ollen kuhunkin kuvaan tuli laittaa teksti, jossa kehoitetaan klikkaamaan kuvaa.

Tämän banneripohjan toteuttamisessa hyödynsin Edgehero-nimistä JavaScript-kirjastoa. Kyseinen JavaScript-kirjasto otettiin käyttöön samalla tavoin kuten haitaribannerissakin, eli CDN-linkin avulla. Tässä projektissa kahdelle elementille lisättiin class-määrytykset, jotta projekti osaisi hyödyntää ulkopuolista JavaScript-tiedostoa oikein.

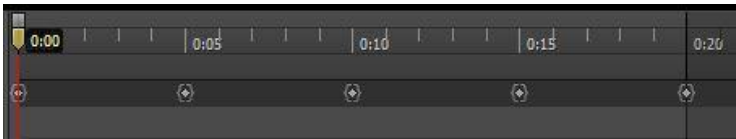
Projektissa (kuvio 20) on div-elementti nimeltään Container, jolle lisäsin class-määritelmän "perspective_3000px center". Container-divin sisällä on div-elementti Cube, jolle lisäsin class-määritelmän "cube3d". Cube-elementin sisällä on kuusi div-elementtiä, jotka ovat kuution jokainen sivu. Kuutioon tarvittiin vain kuution neljää sivua, koska kuution haluttiin pyörähtävän vain x-akselinsa ympäri. Siitä huolimatta tarvittiin kaikki kuusi sivua, jotta kuutio toimisi halutulla tavalla. Kuution pohjan sekä päällä olevan kuvan tulee olla ylimmäisinä, ne toteutettiin tyhjinä div-elementteinä. Muiden kuvien osalta järjestys on siten, että kuution ensimmäinen kuva on alimmaisena, toinen toiseksi alimmaisena ja niin edelleen (kuvio 20).



KUVIO 20. Näkymä elements-paneelistä, jossa näkyy kuutiokuvien oikeaoppinen järjestys

Pyyhkäisytoiminto vastaa klikkaustoimintoa, joten linkkitoiminnallisuuden toteutumiseksi toteutettiin kaksi kuutiopohjaa, joista toinen oli itsestään pyörähtävä ja toinen pyyhkäisytoiminnolla pyörähtävä. Linkkitoiminto toteutettiin itsestään pyörähtävässä kuutiossa suoraan kuvaan. Pyyhkäisykuutioon tehtiin kaikille kuution kuville yksi yhteinen, näkymätön linkkinappi klikkauskehote-tekstin päälle. Pyyhkäisytoiminnallisessa kuutiossa on siis vain yksi linkkiosoite.

Yksinkertaisuudessaan kuution toiminnallisuus perustuu Cube-elementin pyörähdykseen. Kummassakin kuutiossa Cube-elementille määriteltiin samat toiminnallisuudet, mutta eri paikkaan. Itsestään pyörähtävässä kuutiossa toiminnallisuudet määriteltiin aikajanalle laukaisimien (trigger) avulla (kuvio 21). Triggerin saa lisättyä siirtämällä playhead aikajanalla haluttuun kohtaan, näppäinkomennolla Ctrl+T.



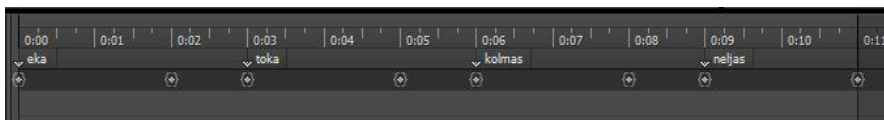
KUVIO 21. Aikajanalle on määritelty tapahtuman laukaisimet, eli triggerit

Kuution haluttiin pyörähtävän kolmen sekunnin välein. Alussa kuution kääntöaste on 0 astetta. Playheadia siirrettiin aikajanalla viiden sekunnin kohdalle ja sille tehtiin määritys pyörähtää 90 astetta vastapäivää, eli -90 astetta ja käännös määriteltiin kestämään kaksi sekuntia (kuvio 22). Muihin laukaisimiin määrittelin täysin samat pyörähdysasetukset, mutta lisäsin kääntökulmaan aina -90 astetta. Viimeiseen laukaisimeen asetin loop-asetuksen, "play(0)", joka aloittaa animaation alusta, eli kuutio pyörähtää takaisin alkuun.



KUVIO 22. Viiden sekunnin kohdalla kuutio pyörähtää -90 astetta. Kuution pyörähdys kestää kaksi sekuntia

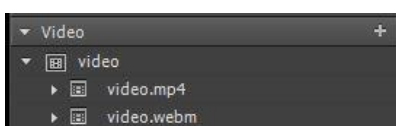
Pyyhkäisytoiminnolla pyörähtävä kuutio eroaa itsestään pyörähtävästä siten, että toiminnallisuus määriteltiin aikajanelle sekä laukaisimia että kylltejä hyödyntäen. Pyyhkäistävässä kuutiossa on samat laukaisin-asetukset kuin itsestään pyörähtävässä kuutiossa, mutta jokaisen pyörähdysten jälkeen aikajana pysäytettiin laukaisimilla. Lisäksi aikajanelle, jokaisen pyörähdysten alkuun, määriteltiin kylltit, jotka nimettiin (kuvio 23). Pyyhkäisytoiminnallisuuden tuli toimia niin oikealle kuin vasemmalle, joten kuhunkin kuution kuvaan määriteltiin toiminnan laukaisimiksi swipeleft sekä swiperight. Toiminnallisuudeksi määriteltiin toistaa animaatio seuraavan tai edellisen kylltin kohdalla.



KUVIO 23. Näkymä pyyhkäistävän kuution aikajanelta

5.4 Videobanneri

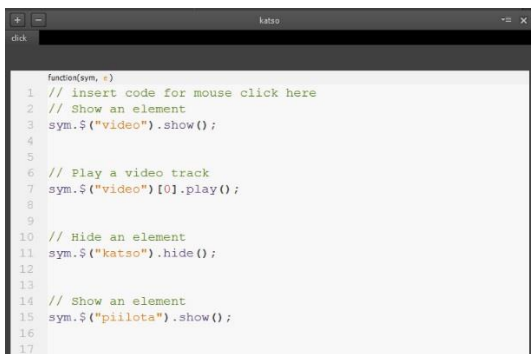
Videobannerin (liite 4) toteutus oli varsin yksinkertaista. Bannerin toteutettiin paraatikokoiseen banneripohjaan, jotta video pääsisi paremmin oikeuksiinsa. Videobanneria varten tarvittiin kaksi videotiedostoa, joista toinen on mp4- ja toinen webm-muodossa. Jotta nämä kaksi tiedostoa ryhmittyvät projektiin oikein, ne täytyy nimetä identtisesti. Videot tuodaan projektiin Import-komennolla ja ne ilmestyvät Library-paneelin video-osioon (kuvio 24).



KUVIO 24. Identtisesti nimetyt videotiedostot ryhmittyvät Video-osioon automaattisesti

Videotiedosto, joka siis sisältää nämä molemmat tiedostomuodot, siirretään Library-paneelistä Stagelle haluttuun kohtaan. Videotiedostolle on omat asetuksensa Properties-paneelissa, kuten soitetaanko video heti, kun projekti on latautunut (autoplay) tai halutaanko videolle omat hallintatyökalut (controls). Kuten luvussa 4 mainittiin, IAB suosittelee, että video alkaisi vasta käyttäjän toimesta. Koska tätä kyseistä videota hallitaan nappia painamalla, videolle ei asetettu autoplay- eikä controls-määrittelyksiä. Videolle asetettiin myös Preloader-asetus, koska banneri on videon vuoksi suhteellisen raskas ja näin ollen latautuu hitaammin kuin perinteiset bannerit.

Päällisin puolin videobanneri näyttää miltä tahansa paraati-kokoiselta bannerilta, jossa on tekstiä ja kuvaa. Video piilotettiin bannerin sisälle display-arvolla "off". Näin ollen video ilmestyy näkyviin vasta sitten kun käyttäjä painaa "Katso video"-nappia. Napin painalluksella videon saa halutessaan myös pois. Banneriin toteutettiin päällekkäin kaksi nappia, Katso video ja Piilota video. Kun käyttäjä painaa Katso video-nappia, niin nappi muuttuu Piilota video -napiksi ja toisin päin. Tämä toteutettiin jQueryn hide- ja show-komennoilla. Kuviosta 25 näkyy esimerkkinä Katso video -napille määritellyt toiminnallisuudet.



KUVIO 25. Katso-napille määritellyt hiiren klikkaus -toiminnallisuudet

5.5 Aineistonsiirto

Luvussa 4.1 kerrottiin HTML5-aineiston siirtämisestä Cxense-ohjelmistoon. Aineiston siirtämistä suoraan Cxense-ohjelmistoon tutkittiin ja kokeiltiin, mutta varsin pian huomattiin, että se on työnkulun kannalta varsin monimutkaista. Näin ollen aineistot siirrettiin Cxense-ohjelmistoon kuten aiemminkin, eli pilvessä toimivan tiedostonhallintaohjelman kautta. Tämä tuo jonkin verran lisätyötä tuotannolle, mutta tällä hetkellä se on ainoa varteenotettava vaihtoehto aineistonsiirtoon.

6 TULOKSET

Opinnäytetyön kehittämistehtävänä oli selvittää, kuinka Adobe Edge Animate soveltuu lehtitalon tuleviin tarpeisiin ja kuinka se sulautuu jo olemassa oleviin järjestelmiin. Ohjelmalla on todella helppo tehdä näyttävän näköisiä display-mainoksia ilman ohjelmointitaitoja, sillä ohjelma rakentaa kaiken ohjelmakoodin taustalla. Ohjelman edistyneempi käyttö sen sijaan vaatii ulkoisia JavaScript-kirjastoja, jotka kirjastosta riippuen edellyttävät enemmän tai vähemmän ymmärrystä ohjelmoinnista ja ohjelman käyttämistä web-teknologioista.

Opinnäytetyön kehitystehtävien tuloksena syntyi viisi banneripohjaa, jota Kalevan ilmoitustuotanto voi käyttää markkinoinnillisiin tarpeisiin hieman muokattuna. Banneripohjista pyrittiin tekemään mahdollisimman helposti muokattavat. Käytännössä banneripohjiin tarvitsee vaihtaa vain halutut kuvat ja tekstit. Muutamassa banneripohjassa käytettiin ulkoista JavaScript-kirjastoa ohjelman omien toiminnallisuuksien tukena, joten ohjelmakoodiin koskemiselta tuotanto ei banneripohjia hyödyntäessään voi valitettavasti vältyä. Tuotannolle laadittiin kuhunkin banneripohjaan selkeät työnkuluohjeistukset, joiden toivotaan auttavan banneripohjien hyödyntämistä tulevaisuudessa.

Päänvaivaa aiheutti eniten selaintuki, eritoten Internet Explorerin selaintuki. Uusin IE:n versio ei esimerkiksi tue vielä kunnolla CSS3-tyyliohjekielen 3D-määrittäjiä. Jotta puutteellisen selaintuen aiheuttamalta turhautumiselta vältyttäisiin, suositellaan käyttämään bannereissa poster-toimintoa.

Aineistonsiirron suhteen pitäyduttiin samassa aineistonsiirtomenetelmässä kuin ennen opinnäytetyön aloittamista. Nykyinen aineistonsiirtomenettely vaatii tuotannolta yhden työvaiheen lisää aikaisempaan aineistonsiirtoon verrattuna, mutta se oli selvittelyn jälkeen paras, jollei ainoa vaihtoehto tarjolla olleista. Aineistonsiirrosta uskotaan tulevan luonnollinen osa bannerituotantoa, kunhan aineistosiirtokäytäntö automatisoituu.

Yhteenvetona voidaan todeta Adobe Edge Animate -ohjelman soveltuvan mainiosti lehtitalon tuleviin tarpeisiin, eli HTML5-bannereiden toteuttamiseen, kunhan aineistonsiirtomenetelmä saadaan sel-laiseksi, ettei se vaadi ylimääräisiä työvaiheita tuotannolta ja selainten tuki saadaan oikealla tasolla.

7 POHDINTA

Koen suurena kunniana saadessani mahdollisuuden tehdä opinnäytetyöni toimeksiantajalle, joka on yksi Suomen suurimpia sanomalehtiä. Hienoa on myös se, että Kalevalla on osaamista tuottaa itse osa Kalevan verkkosivustolle tulevasta display-sisällöstä. Otin haasteen innolla vastaan, vaikka lähdin liikkeelle nollasta. En tiennyt ohjelmasta kuin nimen ja sen käyttämät teknologiat. Katsoin ja luin lukuisia tutoriaaleja liittyen Animateen ja sen käyttämiin teknologioihin. Muutaman kehitystehtävän kohdalla tuli rapsuteltua päätä ankarastikin, mutta alkuasetelmiin suhteutettuna onnistuin mielestäni vähintään hyvin. Opin ohjelman sujuvan käytön huomaamattani kehitystehtäviä toteuttaessani.

Mielestäni Adobe Edge Animate antaa valtavasti mahdollisuuksia luoda uutta internetin ihmeelliseen maailmaan riippumatta käyttäjän ohjelmointitaidoista. Ohjelmointitaitoiselle Animaten tarjoamat mahdollisuudet ovat lähes rajattomat. Herää kuitenkin kysymys, tuleeko tulevaisuuden bannerituotanto olemaan osa webkehityksen osa-aluetta vai täytyykö printtimainonnan graafikolla kehittää osaamistaan vielä modernien web-teknologioidenkin osalta.

Onneksi on olemassa ohjelmisto, joka hoitaa ohjelmoinnin graafikon puolesta. Minusta Edge Animaten hienous onkin juuri siinä, että se luo ohjelmakoodin huomaamatta työskentelyn taustalla. Monet Animaten toimintaa ja työnkulkua helpottavat JavaScript-kirjastot on helppo ottaa käyttöön, vaikka ei ymmärtäisi ohjelmoinnista mitään. Animate hyödyntää toiminnassaan runsaasti JavaScript-kieltä, mutta mielestäni se hyödyntää sitä älykkäästi. Ohjelma sisältää myös paljon automatisoituja toimintoja kuten kansiorakenteen luomisen.

Päänvaivaa kehitystehtäviä tehdessä tuottivat selaintuki, eritoten Internet Explorer sekä Animaten ja Kalevan verkkosivuston välissä olevat järjestelmät. Tuotannon kannalta aineistonsiirtoon on saatava sellainen ratkaisu, joka ei tuo yhtään lisätyötä verrattuna edelliseen käytäntöön. Edistyneempää bannerituotantoa ajatellen, suosittelen toimeksiantajaa tutustumaan myös moderneja web-teknologioita hyödyntäviin verkkomainonnan ohjelmistoihin, joista saa lisätukea Animatella toteutettavan bannerituotannon rinnalle. Opinnäytetyöni oli vain pintaraapaisu ohjelman tarjoamista mahdollisuuksista. Kaiken kaikkiaan Animate on älykäs ohjelma. Suosittelen lämpimästi, sillä HTML5 on jo täällä.

LÄHTEET

DigiKaleva. 2014a. Viitattu 17.9.2014, <http://digi.kaleva.fi/tuotteet/mediatila/kaleva.fi.html>

DigiKaleva. 2014b. Aineisto-ohjeet. Viitattu 17.9.2014, <http://digi.kaleva.fi/aineisto-ohjeet.html>
17.9.2014

Edgehero. 2014. Viitattu 11.10.2014, <http://www.edgehero.com/edgehero>

Edge Docks. 2014a. Viitattu 29.8.2014, <http://www.edgedocks.com/content/2013/12/understanding-file-structure-animate-project>

Edge Docks. 2014b. Viitattu 15.7.2014, <http://www.edgedocks.com/edgecommons>

Emediate Classroom. 2014. HTML5 - Hints and best practice. Viitattu 16.10.2014, http://classroom.emediate.com/doku.php/faq:best_practice_html5

Greensock. 2014. Viitattu 3.9.2014, <http://greensock.com/get-started-js>

Haataja, J., kehityspäällikkö, Kaleva, digitaalinen liiketoiminta. 2014. Haastattelu. 11.9.2014. Tekijän hallussa.

Haataja, J., kehityspäällikkö, Kaleva. Selaimet. Sähköpostiviesti jukka.haataja@kaleva.fi, eliisa.niskanen@kaleva.fi 06.11.2014

IAB Finland 2014a. Viitattu 26.10.2014, <http://www.iab.fi/iab-finland.html>

IAB Finland 2014b. Viitattu 5.9.2014, <http://www.iab.fi/digimainonnan-abc/suosituksset-2.html>

IAB Finland 2014c. Viitattu 5.9.2014, <http://www.iab.fi/media/pdf-tiedostot/standardit-ja-oppaat/iab-finlandin-suositus-paraatipaikka.pdf>

IAB Finland 2014d. Verkkomainonta vahvassa kasvussa ensimmäisellä vuosineljänneksellä. Viitattu 16.9.2014, <http://www.iab.fi/ajax/ajankohtaista-kaikki/uutiset/uutiset/verkkomainonta-vahvassa-kasvussa-ensimmaisella-vuosineljanneksella.html>

jQuery. 2014. Viitattu 17.9.2014, <http://jquery.com>

Järvilehto, T. 2012. Bannerit eli display-mainonta. Teoksessa Häivälä, J. & Paloheimo, T., Klikkaa tästä – Internetmarkkinoinnin käsikirja 2.0. Vaasa: Mainostajien liitto, 96–103.

Konsepto. 2014. Viitattu 5.10.2014, <http://konsepto.fi/palvelut/toteutus-ja-testaus>

Korpela, J. 2013. CSS3 – uudet mahdollisuudet. Jyväskylä. Docendo.

Korpela, J. 2014. HTML5 käsikirja. Jyväskylä. Docendo.

KWD Digital. 2014a. Viitattu 5.9.2014, <http://www.kwd.fi/sanasto/banneri>

KWD Digital. 2014b. Viitattu 13.9.2014, <http://www.kwd.fi/palvelut/digitaalinen-markkinointi/banner-imainonta>

Labrecque, J. 2012. Learning Adobe Edge Animate. Birmingham-Mumbai. Packt Publishing

Mediareaktori. 2014. Sanasto. Viitattu 17.9.2014, <http://www.mediareaktori.fi/sanasto.php>

Peltokorpi, S., tuotantoyksikön päällikkö, Kaleva. Bannerien määrät. Sähköpostiviesti sami.peltokorpi@kaleva.fi, eliisa.niskanen@kaleva.fi 17.9.2014

Rohde, M. 2013. Adobe Edge Animate CC for Dummies. Kanada. John Wiley & Sons, Inc.

TNS Metrix. 2014. Suomen web-sivustojen viikkoluvut. Viitattu 7.11.2014. <http://tnsmetrix.tns-gallup.fi/public/>

W3Schools. 2014a. Viitattu 17.9.2014, http://www.w3schools.com/jquery/jquery_intro.asp

W3Schools. 2014b. Viitattu 28.9.2014, http://www.w3schools.com/jquery/jquery_syntax.asp

W3Schools. 2014c. Viitattu 26.10.2014, http://www.w3schools.com/jquery/jquery_selectors.asp

W3Schools. 2014d. Viitattu 12.10.2014, http://www.w3schools.com/html/html5_video.asp

W3Schools. 2014e. Viitattu 2.9.2014, http://www.w3schools.com/css/css3_2dtransforms.asp

Wikipedia. 2014a. Cascading Style Sheets. Viitattu 3.5.2014, http://en.wikipedia.org/wiki/cascading_style_sheets#css_3

Wikipedia. 2014b. Viitattu 11.7.2014, http://en.wikipedia.org/wiki/Adobe_Edge

Williams, T. 2013. Viitattu 5.10.2014, The difference between responsive and adaptive web design. <http://www.responsys.com/blogs/nsm/cross-channel-marketing/difference-responsive-adaptive-web-design/>





Visit Oulu.fi

Oulun seudun tapahtumakalenteri

Löydä parhaat menot ja tapahtumat!

Tapahtumakalenteriin

www.kaleva.fi/visitoulu

KALEVA.fi



Visit Oulu.fi

Aktiviteetteja, kulttuuria ja seutukohteita

Oulu on täynnä elämämyksiä

Katso parhaat vinkit ja kohteet!

www.kaleva.fi/visitoulu

KALEVA.fi



Oulu Capital of Northern Scandinavia

Ajankohtaisia kuulumisia Oulun business-elämästä ja asumisesta.

Lue hyviä uutisia Oulusta!

www.kaleva.fi/oulusta-kuuluu

KALEVA.fi



Oulu Capital of Northern Scandinavia

100 tekoa Oulusta

Pieniä ja suuria tekoja kohti parempaa Oulua.

Katso uusin valinta!

www.kaleva.fi/oulusta-kuuluu

KALEVA.fi

3D-KUUTIOKARUSELLI (ITSESTÄÄN PYÖRÄHTÄVÄ)

LIITE 3

Kivi kynttilälyhty
väri
pomeranssi
36,90



Katso tästä
lisätietoja >>



Kivi kynttilälyhty
väri
pomeranssi
36,90



Katso tästä
lisätietoja >>



Kivi kynttilälyhty
väri
kirkas matta
25,60



Katso tästä
lisätietoja >>



Kivi kynttilälyhty
väri
pomeranssi
36,90



Katso tästä
lisätietoja >>



Kivi kynttilälyhty
väri
kirkas matta
25,60



Katso tästä
lisätietoja >>



Kivi kynttilälyhty
väri
kirkas matta
25,60



Katso tästä
lisätietoja >>



Kivi kynttilälyhty
väri
kirkas matta
25,60



Katso tästä
lisätietoja >>



KALEVA

muuttuu tabloidiksi
14.2.2015

Alla olevasta linkistä voit
katsoa, videon intron

Katso video tästä

- pienempi ja kätevämpi koko
- enemmän sivuja
- paljon sarjakuvia

Sama Kaleva, mutta
eri koossa



KALEVA

muuttuu tabloidiksi
14.2.2015

Alla olevasta linkistä voit
katsoa, videon intron

Pilota video

UUSI TABLOID-KALEVA - ON SE KÄTEVÄ...

